DSP56366 24-Bit Digital Signal Processor

User Manual

Document Number: DSP56366UM
Rev. 4
08/2006

freescale™
semiconductor

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

***For Literature Requests Only*:**
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-521-6274 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2006. All rights reserved.

Contents

1 DSP56366 Overview 1-1

1.1 Introduction 1-1
1.2 DSP56300 Core Description 1-2
1.3 DSP56366 Audio Processor Architecture 1-3
1.4 DSP56300 Core Functional Blocks 1-3
1.4.1 Data ALU 1-4
1.4.1.1 Data ALU Registers 1-4
1.4.1.2 Multiplier-Accumulator (MAC) 1-4
1.4.2 Address Generation Unit (AGU) 1-5
1.4.3 Program Control Unit (PCU) 1-5
1.4.4 Internal Buses 1-6
1.4.5 Direct Memory Access (DMA) 1-6
1.4.6 PLL-based Clock Oscillator 1-7
1.4.7 JTAG TAP and OnCE Module 1-7
1.4.8 On-Chip Memory 1-7
1.4.9 Off-Chip Memory Expansion 1-8
1.5 Peripheral Overview 1-8
1.5.1 Host Interface (HDI08) 1-9
1.5.2 General Purpose Input/Output (GPIO) 1-9
1.5.3 Triple Timer (TEC) 1-9
1.5.4 Enhanced Serial Audio Interface (ESAI) 1-10
1.5.5 Enhanced Serial Audio Interface 1 (ESAI_1) 1-10
1.5.6 Serial Host Interface (SHI) 1-10
1.5.7 Digital Audio Transmitter (DAX) 1-10

2 Signal/Connection Descriptions 2-1

2.1 Signal Groupings 2-1
2.2 Power 2-3
2.3 Ground 2-3
2.4 Clock and PLL 2-4
2.5 External Memory Expansion Port (Port A) 2-5
2.5.1 External Address Bus 2-5
2.5.2 External Data Bus 2-5
2.5.3 External Bus Control 2-5
2.6 Interrupt and Mode Control 2-7
2.7 PARALLEL HOST INTERFACE (HDI08) 2-9
2.8 Serial Host Interface 2-12
2.9 Enhanced Serial Audio Interface 2-15
2.10 Enhanced Serial Audio Interface_1 2-19
2.11 SPDIF Transmitter Digital Audio Interface 2-21
2.12 Timer 2-22
2.13 JTAG/OnCE Interface 2-22

3 Memory Configuration 3-1

3.1 Data and Program Memory Maps . . . 3-1
3.1.1 Reserved Memory Spaces . . . 3-11
3.1.2 Program ROM Area Reserved for Motorola Use . . . 3-11
3.1.3 Bootstrap ROM . . . 3-11
3.1.4 Dynamic Memory Configuration Switching . . . 3-11
3.1.5 External Memory Support . . . 3-12
3.2 Internal I/O Memory Map . . . 3-12

4 Core Configuration . . . 4-1

4.1 Introduction . . . 4-1
4.2 Operating Mode Register (OMR) . . . 4-1
4.2.1 Asynchronous Bus Arbitration Enable (ABE) - Bit 13 . . . 4-2
4.2.2 Address Attribute Priority Disable (APD) - Bit 14 . . . 4-2
4.2.3 Address Tracing Enable (ATE) - Bit 15 . . . 4-2
4.2.4 Patch Enable (PEN) - Bit 23 . . . 4-2
4.3 Operating Modes . . . 4-4
4.4 Interrupt Priority Registers . . . 4-6
4.5 DMA Request Sources . . . 4-12
4.6 PLL Initialization . . . 4-13
4.6.1 PLL Multiplication Factor (MF0-MF11) . . . 4-13
4.6.2 PLL Pre-Divider Factor (PD0-PD3) . . . 4-13
4.6.3 Crystal Range Bit (XTLR) . . . 4-13
4.6.4 XTAL Disable Bit (XTLD) . . . 4-13
4.7 Device Identification (ID) Register . . . 4-13
4.8 JTAG Identification (ID) Register . . . 4-13
4.9 JTAG Boundary Scan Register (BSR) . . . 4-14

5 General Purpose Input/Output . . . 5-1

5.1 Introduction . . . 5-1
5.2 Programming Model . . . 5-1
5.2.1 Port B Signals and Registers . . . 5-1
5.2.2 Port C Signals and Registers . . . 5-1
5.2.3 Port D Signals and Registers . . . 5-1
5.2.4 Port E Signals and Registers . . . 5-2
5.2.5 Timer/Event Counter Signals . . . 5-2

6 Host Interface (HDI08) . . . 6-1

6.1 Introduction . . . 6-1
6.2 HDI08 Features . . . 6-1
6.2.1 Interface - DSP side . . . 6-1
6.2.2 Interface - Host Side . . . 6-2
6.3 HDI08 Host Port Signals . . . 6-3
6.4 HDI08 Block Diagram . . . 6-4
6.5 HDI08 – DSP-Side Programmer’s Model . . . 6-5

6.5.1 Host Receive Data Register (HORX) 6-6
6.5.2 Host Transmit Data Register (HOTX) 6-6
6.5.3 Host Control Register (HCR) 6-7
6.5.3.1 HCR Host Receive Interrupt Enable (HRIE) Bit 0 6-7
6.5.3.2 HCR Host Transmit Interrupt Enable (HTIE) Bit 1 6-7
6.5.3.3 HCR Host Command Interrupt Enable (HCIE) Bit 2 6-7
6.5.3.4 HCR Host Flags 2,3 (HF2,HF3) Bits 3-4 6-8
6.5.3.5 HCR Host DMA Mode Control Bits (HDM0, HDM1, HDM2) Bits 5-7 6-8
6.5.3.6 HCR Reserved Bits 8-15 6-10
6.5.4 Host Status Register (HSR) 6-10
6.5.4.1 HSR Host Receive Data Full (HRDF) Bit 0 6-10
6.5.4.2 HSR Host Transmit Data Empty (HTDE) Bit 1 6-10
6.5.4.3 HSR Host Command Pending (HCP) Bit 2 6-10
6.5.4.4 HSR Host Flags 0,1 (HF0,HF1) Bits 3-4 6-11
6.5.4.5 HSR Reserved Bits 5-6, 8-15 6-11
6.5.4.6 HSR DMA Status (DMA) Bit 7 6-11
6.5.5 Host Base Address Register (HBAR) 6-11
6.5.5.1 HBAR Base Address (BA[10:3]) Bits 0-7 6-11
6.5.5.2 HBAR Reserved Bits 8-15 6-12
6.5.6 Host Port Control Register (HPCR) 6-12
6.5.6.1 HPCR Host GPIO Port Enable (HGEN) Bit 0 6-12
6.5.6.2 HPCR Host Address Line 8 Enable (HA8EN) Bit 1 6-13
6.5.6.3 HPCR Host Address Line 9 Enable (HA9EN) Bit 2 6-13
6.5.6.4 HPCR Host Chip Select Enable (HCSEN) Bit 3 6-13
6.5.6.5 HPCR Host Request Enable (HREN) Bit 4 6-13
6.5.6.6 HPCR Host Acknowledge Enable (HAEN) Bit 5 6-13
6.5.6.7 HPCR Host Enable (HEN) Bit 6 6-13
6.5.6.8 HPCR Reserved Bit 7 6-13
6.5.6.9 HPCR Host Request Open Drain (HROD) Bit 8 6-14
6.5.6.10 HPCR Host Data Strobe Polarity (HDSP) Bit 9 6-14
6.5.6.11 HPCR Host Address Strobe Polarity (HASP) Bit 10 6-14
6.5.6.12 HPCR Host Multiplexed bus (HMUX) Bit 11 6-14
6.5.6.13 HPCR Host Dual Data Strobe (HDDS) Bit 12 6-14
6.5.6.14 HPCR Host Chip Select Polarity (HCSP) Bit 13 6-15
6.5.6.15 HPCR Host Request Polarity (HRP) Bit 14 6-15
6.5.6.16 HPCR Host Acknowledge Polarity (HAP) Bit 15 6-15
6.5.7 Data direction register (HDDR) 6-15
6.5.8 Host Data Register (HDR) 6-16
6.5.9 DSP-Side Registers After Reset 6-16
6.5.10 Host Interface DSP Core Interrupts 6-17
6.6 HDI08 – External Host Programmer's Model 6-18
6.6.1 Interface Control Register (ICR) 6-19
6.6.1.1 ICR Receive Request Enable (RREQ) Bit 0 6-20
6.6.1.2 ICR Transmit Request Enable (TREQ) Bit 1 6-20
6.6.1.3 ICR Double Host Request (HDRQ) Bit 2 6-21

6.6.1.4 ICR Host Flag 0 (HF0) Bit 3 6-21
6.6.1.5 ICR Host Flag 1 (HF1) Bit 4 6-21
6.6.1.6 ICR Host Little Endian (HLEND) Bit 5 6-22
6.6.1.7 ICR Initialize Bit (INIT) Bit 7 6-23
6.6.2 Command Vector Register (CVR) 6-23
6.6.2.1 CVR Host Vector (HV[6:0]) Bits 0–6 6-23
6.6.2.2 CVR Host Command Bit (HC) Bit 7 6-24
6.6.3 Interface Status Register (ISR) 6-24
6.6.3.1 ISR Receive Data Register Full (RXDF) Bit 0 6-24
6.6.3.2 ISR Transmit Data Register Empty (TXDE) Bit 1 6-24
6.6.3.3 ISR Transmitter Ready (TRDY) Bit 2 6-25
6.6.3.4 ISR Host Flag 2 (HF2) Bit 3 6-25
6.6.3.5 ISR Host Flag 3 (HF3) Bit 4 6-25
6.6.3.6 ISR Reserved Bits 5-6 6-25
6.6.3.7 ISR Host Request (HREQ) Bit 7 6-25
6.6.4 Interrupt Vector Register (IVR) 6-26
6.6.5 Receive Byte Registers (RXH:RXM:RXL) 6-26
6.6.6 Transmit Byte Registers (TXH:TXM:TXL) 6-26
6.6.7 Host Side Registers After Reset 6-27
6.6.8 General Purpose INPUT/OUTPUT (GPIO) 6-27
6.7 Servicing The Host Interface 6-28
6.7.1 HDI08 Host Processor Data Transfer 6-28
6.7.2 Polling 6-28
6.7.3 Servicing Interrupts 6-29

7 Serial Host Interface 7-1

7.1 Introduction 7-1
7.2 Serial Host Interface Internal Architecture 7-2
7.3 SHI Clock Generator 7-2
7.4 Serial Host Interface Programming Model 7-3
7.4.1 SHI Input/Output Shift Register (IOSR)—Host Side 7-5
7.4.2 SHI Host Transmit Data Register (HTX)—DSP Side 7-6
7.4.3 SHI Host Receive Data FIFO (HRX)—DSP Side 7-6
7.4.4 SHI Slave Address Register (HSAR)—DSP Side 7-6
7.4.4.1 HSAR Reserved Bits—Bits 19, 17–0 7-7
7.4.4.2 HSAR I^2C Slave Address (HA[6:3], HA1)—Bits 23–20,18 7-7
7.4.5 SHI Clock Control Register (HCKR)—DSP Side 7-7
7.4.5.1 Clock Phase and Polarity (CPHA and CPOL)—Bits 1–0 7-7
7.4.5.2 HCKR Prescaler Rate Select (HRS)—Bit 2 7-9
7.4.5.3 HCKR Divider Modulus Select (HDM[7:0])—Bits 10–3 7-9
7.4.5.4 HCKR Reserved Bits—Bits 23–14, 11 7-9
7.4.5.5 HCKR Filter Mode (HFM[1:0]) — Bits 13–12 7-9
7.4.6 SHI Control/Status Register (HCSR)—DSP Side 7-10
7.4.6.1 HCSR Host Enable (HEN)—Bit 0 7-10
7.4.6.1.1 SHI Individual Reset 7-11

7.4.6.2 HCSR I2C/SPI Selection (HI2C)—Bit 1 7-11
7.4.6.3 HCSR Serial Host Interface Mode (HM[1:0])—Bits 3–2 7-11
7.4.6.4 HCSR I2C Clock Freeze (HCKFR)—Bit 4 7-11
7.4.6.5 HCSR FIFO-Enable Control (HFIFO)—Bit 5 7-12
7.4.6.6 HCSR Master Mode (HMST)—Bit 6 7-12
7.4.6.7 HCSR Host-Request Enable (HRQE[1:0])—Bits 8–7 7-12
7.4.6.8 HCSR Idle (HIDLE)—Bit 9 7-13
7.4.6.9 HCSR Bus-Error Interrupt Enable (HBIE)—Bit 10 7-13
7.4.6.10 HCSR Transmit-Interrupt Enable (HTIE)—Bit 11 7-13
7.4.6.11 HCSR Receive Interrupt Enable (HRIE[1:0])—Bits 13–12 7-14
7.4.6.12 HCSR Host Transmit Underrun Error (HTUE)—Bit 14 7-14
7.4.6.13 HCSR Host Transmit Data Empty (HTDE)—Bit 15 7-15
7.4.6.14 HCSR Reserved Bits—Bits 23, 18 and 16 7-15
7.4.6.15 Host Receive FIFO Not Empty (HRNE)—Bit 17 7-15
7.4.6.16 Host Receive FIFO Full (HRFF)—Bit 19 7-15
7.4.6.17 Host Receive Overrun Error (HROE)—Bit 20 7-15
7.4.6.18 Host Bus Error (HBER)—Bit 21 7-16
7.4.6.19 HCSR Host Busy (HBUSY)—Bit 22 7-16
7.5 Characteristics Of The SPI Bus 7-16
7.6 Characteristics Of The I2C Bus 7-16
7.6.1 Overview 7-17
7.6.2 I2C Data Transfer Formats 7-18
7.7 SHI Programming Considerations 7-19
7.7.1 SPI Slave Mode 7-19
7.7.2 SPI Master Mode 7-20
7.7.3 I2C Slave Mode 7-21
7.7.3.1 Receive Data in I2C Slave Mode 7-21
7.7.3.2 Transmit Data In I2C Slave Mode 7-22
7.7.4 I2C Master Mode 7-23
7.7.4.1 Receive Data in I2C Master Mode 7-24
7.7.4.2 Transmit Data In I2C Master Mode 7-24
7.7.5 SHI Operation During DSP Stop 7-25

8 Enhanced Serial AUDIO Interface (ESAI) 8-1

8..1 Introduction 8-1
8.2 ESAI Data and Control Pins 8-3
8.2.1 Serial Transmit 0 Data Pin (SDO0) 8-3
8.2.2 Serial Transmit 1 Data Pin (SDO1) 8-3
8.2.3 Serial Transmit 2/Receive 3 Data Pin (SDO2/SDI3) 8-3
8.2.4 Serial Transmit 3/Receive 2 Data Pin (SDO3/SDI2) 8-4
8.2.5 Serial Transmit 4/Receive 1 Data Pin (SDO4/SDI1) 8-4
8.2.6 Serial Transmit 5/Receive 0 Data Pin (SDO5/SDI0) 8-4
8.2.7 Receiver Serial Clock (SCKR) V4
8.2.8 Transmitter Serial Clock (SCKT) 8-5
8.2.9 Frame Sync for Receiver (FSR) 8-6

8.2.10 Frame Sync for Transmitter (FST) 8-7
8.2.11 High Frequency Clock for Transmitter (HCKT) 8-7
8.2.12 High Frequency Clock for Receiver (HCKR) 8-7
8.3 ESAI Programming Model 8-7
8.3.1 ESAI Transmitter Clock Control Register (TCCR) 8-8
8.3.1.1 TCCR Transmit Prescale Modulus Select (TPM7–TPM0) - Bits 0–7 8-8
8.3.1.2 TCCR Transmit Prescaler Range (TPSR) - Bit 8 8-9
8.3.1.3 TCCR Tx Frame Rate Divider Control (TDC4–TDC0) - Bits 9–13 8-10
8.3.1.4 TCCR Tx High Frequency Clock Divider (TFP3-TFP0) - Bits 14–17 8-11
8.3.1.5 TCCR Transmit Clock Polarity (TCKP) - Bit 18 8-12
8.3.1.6 TCCR Transmit Frame Sync Polarity (TFSP) - Bit 19 8-12
8.3.1.7 TCCR Transmit High Frequency Clock Polarity (THCKP) - Bit 20 8-12
8.3.1.8 TCCR Transmit Clock Source Direction (TCKD) - Bit 21 8-12
8.3.1.9 TCCR Transmit Frame Sync Signal Direction (TFSD) - Bit 22 8-12
8.3.1.10 TCCR Transmit High Frequency Clock Direction (THCKD) - Bit 23 8-12
8.3.2 ESAI Transmit Control Register (TCR) 8-12
8.3.2.1 TCR ESAI Transmit 0 Enable (TE0) - Bit 0 8-13
8.3.2.2 TCR ESAI Transmit 1 Enable (TE1) - Bit 1 8-13
8.3.2.3 TCR ESAI Transmit 2 Enable (TE2) - Bit 2 8-14
8.3.2.4 TCR ESAI Transmit 3 Enable (TE3) - Bit 3 8-14
8.3.2.5 TCR ESAI Transmit 4 Enable (TE4) - Bit 4 8-14
8.3.2.6 TCR ESAI Transmit 5 Enable (TE5) - Bit 5 8-15
8.3.2.7 TCR Transmit Shift Direction (TSHFD) - Bit 6 8-15
8.3.2.8 TCR Transmit Word Alignment Control (TWA) - Bit 7 8-15
8.3.2.9 TCR Transmit Network Mode Control (TMOD1-TMOD0) - Bits 8-9 8-16
8.3.2.10 TCR Tx Slot and Word Length Select (TSWS4-TSWS0) - Bits 10-14 8-18
8.3.2.11 TCR Transmit Frame Sync Length (TFSL) - Bit 15 8-19
8.3.2.12 TCR Transmit Frame Sync Relative Timing (TFSR) - Bit 16 8-21
8.3.2.13 TCR Transmit Zero Padding Control (PADC) - Bit 17 8-21
8.3.2.14 TCR Reserved Bit - Bits 18 8-21
8.3.2.15 TCR Transmit Section Personal Reset (TPR) - Bit 19 8-21
8.3.2.16 TCR Transmit Exception Interrupt Enable (TEIE) - Bit 20 8-21
8.3.2.17 TCR Transmit Even Slot Data Interrupt Enable (TEDIE) - Bit 21 8-22
8.3.2.18 TCR Transmit Interrupt Enable (TIE) - Bit 22 8-22
8.3.2.19 TCR Transmit Last Slot Interrupt Enable (TLIE) - Bit 23 8-22
8.3.3 ESAI Receive Clock Control Register (RCCR) 8-22
8.3.3.1 RCCR Receiver Prescale Modulus Select (RPM7–RPM0) - Bits 7–0 8-23
8.3.3.2 RCCR Receiver Prescaler Range (RPSR) - Bit 8 8-23
8.3.3.3 RCCR Rx Frame Rate Divider Control (RDC4–RDC0) - Bits 9–13 8-23
8.3.3.4 RCCR Rx High Frequency Clock Divider (RFP3-RFP0) - Bits 14-17 8-23
8.3.3.5 RCCR Receiver Clock Polarity (RCKP) - Bit 18 8-24
8.3.3.6 RCCR Receiver Frame Sync Polarity (RFSP) - Bit 19 8-24
8.3.3.7 RCCR Receiver High Frequency Clock Polarity (RHCKP) - Bit 20 8-24
8.3.3.8 RCCR Receiver Clock Source Direction (RCKD) - Bit 21 8-24
8.3.3.9 RCCR Receiver Frame Sync Signal Direction (RFSD) - Bit 22 8-25

8.3.3.10 RCCR Receiver High Frequency Clock Direction (RHCKD) - Bit 23 8-26
8.3.4 ESAI Receive Control Register (RCR) 8-26
8.3.4.1 RCR ESAI Receiver 0 Enable (RE0) - Bit 0 8-27
8.3.4.2 RCR ESAI Receiver 1 Enable (RE1) - Bit 1 8-27
8.3.4.3 RCR ESAI Receiver 2 Enable (RE2) - Bit 2 8-27
8.3.4.4 RCR ESAI Receiver 3 Enable (RE3) - Bit 3 8-27
8.3.4.5 RCR Reserved Bits - Bits 4-5, 17-18 8-27
8.3.4.6 RCR Receiver Shift Direction (RSHFD) - Bit 6 8-27
8.3.4.7 RCR Receiver Word Alignment Control (RWA) - Bit 7 8-28
8.3.4.8 RCR Receiver Network Mode Control (RMOD1-RMOD0) - Bits 8-9 8-28
8.3.4.9 RCR Receiver Slot and Word Select (RSWS4-RSWS0) - Bits 10-14 8-28
8.3.4.10 RCR Receiver Frame Sync Length (RFSL) - Bit 15 8-30
8.3.4.11 RCR Receiver Frame Sync Relative Timing (RFSR) - Bit 16 8-30
8.3.4.12 RCR Receiver Section Personal Reset (RPR) - Bit 19 8-30
8.3.4.13 RCR Receive Exception Interrupt Enable (REIE) - Bit 20 8-31
8.3.4.14 RCR Receive Even Slot Data Interrupt Enable (REDIE) - Bit 21 8-31
8.3.4.15 RCR Receive Interrupt Enable (RIE) - Bit 22 8-31
8.3.4.16 RCR Receive Last Slot Interrupt Enable (RLIE) - Bit 23 8-31
8.3.5 ESAI Common Control Register (SAICR) 8-31
8.3.5.1 SAICR Serial Output Flag 0 (OF0) - Bit 0 8-32
8.3.5.2 SAICR Serial Output Flag 1 (OF1) - Bit 1 8-32
8.3.5.3 SAICR Serial Output Flag 2 (OF2) - Bit 2 8-32
8.3.5.4 SAICR Reserved Bits - Bits 3-5, 9-23 8-32
8.3.5.5 SAICR Synchronous Mode Selection (SYN) - Bit 6 8-32
8.3.5.6 SAICR Transmit External Buffer Enable (TEBE) - Bit 7 8-33
8.3.5.7 SAICR Alignment Control (ALC) - Bit 8 8-33
8.3.6 ESAI Status Register (SAISR) 8-34
8.3.6.1 SAISR Serial Input Flag 0 (IF0) - Bit 0 8-35
8.3.6.2 SAISR Serial Input Flag 1 (IF1) - Bit 1 8-35
8.3.6.3 SAISR Serial Input Flag 2 (IF2) - Bit 2 8-35
8.3.6.4 SAISR Reserved Bits - Bits 3-5, 11-12, 18-23 8-35
8.3.6.5 SAISR Receive Frame Sync Flag (RFS) - Bit 6 8-35
8.3.6.6 SAISR Receiver Overrun Error Flag (ROE) - Bit 7 8-36
8.3.6.7 SAISR Receive Data Register Full (RDF) - Bit 8 8-36
8.3.6.8 SAISR Receive Even-Data Register Full (REDF) - Bit 9 8-36
8.3.6.9 SAISR Receive Odd-Data Register Full (RODF) - Bit 10 8-36
8.3.6.10 SAISR Transmit Frame Sync Flag (TFS) - Bit 13 8-36
8.3.6.11 SAISR Transmit Underrun Error Flag (TUE) - Bit 14 8-37
8.3.6.12 SAISR Transmit Data Register Empty (TDE) - Bit 15 8-37
8.3.6.13 SAISR Transmit Even-Data Register Empty (TEDE) - Bit 16 8-37
8.3.6.14 SAISR Transmit Odd-Data Register Empty (TODE) - Bit 17 8-37
8.3.7 ESAI Receive Shift Registers 8-40
8.3.8 ESAI Receive Data Registers (RX3, RX2, RX1, RX0) 8-40
8.3.9 ESAI Transmit Shift Registers 8-40
8.3.10 ESAI Transmit Data Registers (TX5, TX4, TX3, TX2,TX1,TX0) 8-40

8.3.11 ESAI Time Slot Register (TSR) 8-40
8.3.12 Transmit Slot Mask Registers (TSMA, TSMB) 8-40
8.3.13 Receive Slot Mask Registers (RSMA, RSMB) 8-42
8.4 Operating Modes 8-43
8.4.1 ESAI After Reset 8-43
8.4.2 ESAI Initialization 8-43
8.4.3 ESAI Interrupt Requests 8-44
8.4.4 Operating Modes – Normal, Network, and On-Demand 8-45
8.4.4.1 Normal/Network/On-Demand Mode Selection 8-45
8.4.4.2 Synchronous/Asynchronous Operating Modes 8-45
8.4.4.3 Frame Sync Selection 8-46
8.4.4.4 Shift Direction Selection 8-46
8.4.5 Serial I/O Flags 8-46
8.5 GPIO - Pins and Registers 8-47
8.5.1 Port C Control Register (PCRC) 8-47
8.5.2 Port C Direction Register (PRRC) 8-47
8.5.3 Port C Data register (PDRC) 8-48
8.6 ESAI Initialization Examples 8-49
8.6.1 Initializing the ESAI Using Individual Reset 8-49
8.6.2 Initializing Just the ESAI Transmitter Section 8-49
8.6.3 Initializing Just the ESAI Receiver Section 8-50

9 Enhanced Serial Audio Interface 1 (ESAI_1) 9-1

9.1 Introduction 9-1
9.2 ESAI_1 Data and Control Pins 9-3
9.2.1 Serial Transmit 0 Data Pin (SDO0_1) 9-3
9.2.2 Serial Transmit 1 Data Pin (SDO1_1) 9-3
9.2.3 Serial Transmit 2/Receive 3 Data Pin (SDO2_1/SDI3_1) 9-3
9.2.4 Serial Transmit 3/Receive 2 Data Pin (SDO3_1/SDI2_1) 9-3
9.2.5 Serial Transmit 4/Receive 1 Data Pin (SDO4_1/SDI1_1) 9-3
9.2.6 Serial Transmit 5/Receive 0 Data Pin (SDO5_1/SDI0_1) 9-4
9.2.7 Receiver Serial Clock (SCKR_1) 9-4
9.2.8 Transmitter Serial Clock (SCKT_1) 9-4
9.2.9 Frame Sync for Receiver (FSR_1) 9-4
9.2.10 Frame Sync for Transmitter (FST_1) 9-4
9.3 ESAI_1 Programming Model 9-4
9.3.1 ESAI_1 Multiplex Control Register (EMUXR) 9-5
9.3.2 ESAI_1 Transmitter Clock Control Register (TCCR_1) 9-5
9.3.2.1 TCCR_1 Tx High Freq. Clock Divider (TFP3-TFP0) - Bits 14–17 9-6
9.3.2.2 TCCR_1 Tx High Freq. Clock Polarity (THCKP) - Bit 20 9-6
9.3.2.3 TCCR_1 Tx High Freq. Clock Direction (THCKD) - Bit 23 9-6
9.3.3 ESAI_1 Transmit Control Register (TCR_1) 9-8
9.3.4 ESAI_1 Receive Clock Control Register (RCCR_1) 9-9
9.3.4.1 RCCR_1 Rx High Freq. Clock Divider (RFP3-RFP0) - Bits 14–17 9-9
9.3.4.2 RCCR_1 Rx High Freq. Clock Polarity (RHCKP) - Bit 20 9-9

9.3.4.3 RCCR_1 Rx High Freq. Clock Direction (RHCKD) - Bit 23 9-9
9.3.5 ESAI_1 Receive Control Register (RCR_1) 9-10
9.3.6 ESAI_1 Common Control Register (SAICR_1) 9-10
9.3.7 ESAI_1 Status Register (SAISR_1) 9-10
9.3.8 ESAI_1 Receive Shift Registers 9-11
9.3.9 ESAI_1 Receive Data Registers 9-11
9.3.10 ESAI_1 Transmit Shift Registers 9-11
9.3.11 ESAI_1 Transmit Data Registers 9-11
9.3.12 ESAI_1 Time Slot Register (TSR_1) 9-12
9.3.13 Transmit Slot Mask Registers (TSMA_1, TSMB_1) 9-12
9.3.14 Receive Slot Mask Registers (RSMA_1, RSMB_1) 9-12
9.4 Operating Modes 9-13
9.4.1 ESAI_1 After Reset 9-13
9.5 GPIO - Pins and Registers 9-13
9.5.1 Port E Control Register (PCRE) 9-13
9.5.2 Port E Direction Register (PRRE) 9-14
9.5.3 Port E Data register (PDRE) 9-14

10 Digital Audio Transmitter 10-1

10.1 Introduction 10-1
10.2 DAX Signals 10-2
10.3 DAX Functional Overview 10-2
10.4 DAX Programming Model 10-3
10.5 DAX Internal Architecture 10-4
10.5.1 DAX Audio Data Register (XADR) 10-5
10.5.2 DAX Audio Data Buffers (XADBUFA / XADBUFB) 10-5
10.5.3 DAX Audio Data Shift Register (XADSR) 10-5
10.5.4 DAX Non-Audio Data Register (XNADR) 10-5
10.5.4.1 DAX Channel A Validity (XVA)—Bit 10 10-5
10.5.4.2 DAX Channel A User Data (XUA)—Bit 11 10-6
10.5.4.3 DAX Channel A Channel Status (XCA)—Bit 12 10-6
10.5.4.4 DAX Channel B Validity (XVB)—Bit 13 10-6
10.5.4.5 DAX Channel B User Data (XUB)—Bit 14 10-6
10.5.4.6 DAX Channel B Channel Status (XCB)—Bit 15 10-6
10.5.4.7 XNADR Reserved Bits—Bits 0-9, 16–23 10-6
10.5.5 DAX Non-Audio Data Buffer (XNADBUF) 10-6
10.5.6 DAX Control Register (XCTR) 10-6
10.5.6.1 Audio Data Register Empty Interrupt Enable (XDIE)—Bit 0 10-7
10.5.6.2 Underrun Error Interrupt Enable (XUIE)—Bit 1 10-7
10.5.6.3 Block Transferred Interrupt Enable (XBIE)—Bit 2 10-7
10.5.6.4 DAX Clock Input Select (XCS[1:0])—Bits 3–4 10-7
10.5.6.5 DAX Start Block (XSB)—Bit 5 10-7
10.5.6.6 XCTR Reserved Bits—Bits 6-23 10-7
10.5.7 DAX Status Register (XSTR) 10-7
10.5.7.1 DAX Audio Data Register Empty (XADE)—Bit 0 10-8

10.5.7.2 DAX Transmit Underrun Error Flag (XAUR)—Bit 1 10-8
10.5.7.3 DAX Block Transfer Flag (XBLK)—Bit 2 10-8
10.5.7.4 XSTR Reserved Bits—Bits 3–23 10-8
10.5.8 DAX Parity Generator (PRTYG) 10-9
10.5.9 DAX Biphase Encoder 10-9
10.5.10 DAX Preamble Generator 10-9
10.5.11 DAX Clock Multiplexer 10-9
10.5.12 DAX State Machine 10-10
10.6 DAX Programming Considerations 10-10
10.6.1 Initiating A Transmit Session 10-10
10.6.2 Audio Data Register Empty Interrupt Handling 10-10
10.6.3 Block Transferred Interrupt Handling 10-11
10.6.4 DAX operation with DMA 10-11
10.6.5 DAX Operation During Stop 10-12
10.7 GPIO (PORT D) - Pins and Registers 10-12
10.7.1 Port D Control Register (PCRD) 10-12
10.7.2 Port D Direction Register (PRRD) 10-13
10.7.3 Port D Data Register (PDRD) 10-14

11 Timer/ Event Counter 11-1

11.1 Introduction 11-1
11.2 Timer/Event Counter Architecture 11-1
11.2.1 Timer/Event Counter Block Diagram 11-1
11.2.2 Individual Timer Block Diagram 11-2
11.3 Timer/Event Counter Programming Model 11-3
11.3.1 Prescaler Counter 11-5
11.3.2 Timer Prescaler Load Register (TPLR) 11-5
11.3.2.1 TPLR Prescaler Preload Value PL[20:0] Bits 20–0 11-5
11.3.2.2 TPLR Prescaler Source PS[1:0] Bits 22-21 11-5
11.3.2.3 TPLR Reserved Bit 23 11-6
11.3.3 Timer Prescaler Count Register (TPCR) 11-6
11.3.3.1 TPCR Prescaler Counter Value PC[20:0] Bits 20–0 11-6
11.3.3.2 TPCR Reserved Bits 23–21 11-6
11.3.4 Timer Control/Status Register (TCSR) 11-6
11.3.4.1 TCSR Timer Enable (TE) Bit 0 11-6
11.3.4.2 TCSR Timer Overflow Interrupt Enable (TOIE) Bit 1 11-7
11.3.4.3 TCSR Timer Compare Interrupt Enable (TCIE) Bit 2 11-7
11.3.4.4 TCSR Timer Control (TC[3:0]) Bits 4–7 11-7
11.3.4.5 TCSR Inverter (INV) Bit 8 11-9
11.3.4.6 TCSR Timer Reload Mode (TRM) Bit 9 11-10
11.3.4.7 TCSR Direction (DIR) Bit 11 11-10
11.3.4.8 TCSR Data Input (DI) Bit 12 11-10
11.3.4.9 TCSR Data Output (DO) Bit 13 11-10
11.3.4.10 TCSR Prescaler Clock Enable (PCE) Bit 15 11-11
11.3.4.11 TCSR Timer Overflow Flag (TOF) Bit 20 11-11

11.3.4.12 TCSR Timer Compare Flag (TCF) Bit 21 11-11
11.3.4.13 TCSR Reserved Bits (Bits 3, 10, 14, 16-19, 22, 23) 11-11
11.3.5 Timer Load Register (TLR) 11-12
11.3.6 Timer Compare Register (TCPR) 11-12
11.3.7 Timer Count Register (TCR) 11-12
11.4 Timer Modes of Operation 11-12
11.4.1 Timer Modes 11-13
11.4.1.1 Timer GPIO (Mode 0) 11-13
11.4.1.2 Timer Pulse (Mode 1) 11-14
11.4.1.3 Timer Toggle (Mode 2) 11-14
11.4.1.4 Timer Event Counter (Mode 3) 11-15
11.4.2 Signal Measurement Modes 11-16
11.4.2.1 Measurement Accuracy 11-16
11.4.2.2 Measurement Input Width (Mode 4) 11-16
11.4.2.3 Measurement Input Period (Mode 5) 11-17
11.4.2.4 Measurement Capture (Mode 6) 11-17
11.4.3 Pulse Width Modulation (PWM, Mode 7) 11-18
11.4.4 Watchdog Modes 11-19
11.4.4.1 Watchdog Pulse (Mode 9) 11-19
11.4.4.2 Watchdog Toggle (Mode 10) 11-20
11.4.5 Reserved Modes 11-20
11.4.6 Special Cases 11-20
11.4.6.1 Timer Behavior during Wait 11-20
11.4.6.2 Timer Behavior during Stop 11-21
11.4.7 DMA Trigger 11-21

Appendix A Bootstrap ROM Contents A-1

A.1 DSP56366 Bootstrap Program A-1

Appendix B Equates B-1

Appendix C JTAG BSDL C-1

Appendix D Programmer's Reference D-1

B.1 Introduction B-1
B.1.1 Peripheral Addresses B-1
B.1.2 Interrupt Addresses B-1
B.1.3 Interrupt Priorities B-1
B.1.4 Host Interface Quick Reference B-1
B.1.5 Programming Sheets B-1
B.2 Internal I/O Memory Map B-1
B.3 Interrupt Vector Addresses B-8

B.4 Interrupt Source Priorities (within an IPL) B-10
B.5 Host Interface—Quick Reference B-12
B.6 Programming Sheets B-15

List of Figures

Figure 1-1 DSP56366 Block Diagram 1-1
Figure 2-1 Signals Identified by Functional Group 2-2
Figure 3-1 Memory Maps for MSW=(X,X), CE=0, MS=0, SC=0 3-3
Figure 3-2 Memory Maps for MSW=(X,X), CE=1, MS=0, SC=0 3-3
Figure 3-3 Memory Maps for MSW=(0,0), CE=0 MS=1, SC=0 3-4
Figure 3-4 Memory Maps for MSW=(0,1), CE=0, MS=1, SC=0 3-4
Figure 3-5 Memory Maps for MSW=(1,0), CE=0, MS=1, SC=0 3-5
Figure 3-6 Memory Maps for MSW=(0,0), CE=1, MS=1, SC=0 3-5
Figure 3-7 Memory Maps for MSW=(0,1), CE=1, MS=1, SC=0 3-6
Figure 3-8 Memory Maps for MSW=(1,0), CE=1, MS=1, SC=0 3-6
Figure 3-9 Memory Maps for MSW=(X,X), CE=0, MS=0, SC=1 3-7
Figure 3-10 Memory Maps for MSW=(X,X), CE=1, MS=0, SC=1 3-7
Figure 3-11 Memory Maps for MSW=(0,0), CE=0, MS=1, SC=1 3-8
Figure 3-12 Memory Maps for MSW=(0,1), CE=0, MS=1, SC=1 3-8
Figure 3-13 Memory Maps for MSW=(1,0), CE=0, MS=1, SC=1 3-9
Figure 3-14 Memory Maps for MSW=(0,0), CE=1, MS=1, SC=1 3-9
Figure 3-15 Memory Maps for MSW=(0,1), CE=1, MS=1, SC=1 3-10
Figure 3-16 Memory Maps for MSW=(1,0), CE=1, MS=1, SC=1 3-10
Figure 4-1 Interrupt Priority Register P 4-7
Figure 4-2 Interrupt Priority Register C 4-7
Figure 6-1 HDI08 Block Diagram 6-5
Figure 6-2 Host Control Register (HCR) (X:$FFFFC2) 6-7
Figure 6-3 Host Status Register (HSR) (X:FFFFC3) 6-10
Figure 6-4 Host Base Address Register (HBAR) (X:$FFFFC5) 6-11
Figure 6-5 Self Chip Select logic 6-12
Figure 6-6 Host Port Control Register (HPCR) (X:$FFFFC4) 6-12
Figure 6-7 Single strobe bus 6-14
Figure 6-8 Dual strobes bus 6-15
Figure 6-9 Host Data Direction Register (HDDR) (X:$FFFFC8) 6-16
Figure 6-10 Host Data Register (HDR) (X:$FFFFC9) 6-16
Figure 6-11 HSR-HCR Operation 6-18
Figure 6-12 Interface Control Register (ICR) 6-20
Figure 6-13 Command Vector Register (CVR) 6-23
Figure 6-14 Interface Status Register (ISR) 6-24
Figure 6-15 Interrupt Vector Register (IVR) 6-26
Figure 6-16 HDI08 Host Request Structure 6-29
Figure 7-1 Serial Host Interface Block Diagram 7-2
Figure 7-2 SHI Clock Generator 7-3
Figure 7-3 SHI Programming Model—Host Side 7-3

Figure 7-4 SHI Programming Model—DSP Side 7-4
Figure 7-5 SHI I/O Shift Register (IOSR) 7-6
Figure 7-6 SPI Data-To-Clock Timing Diagram 7-8
Figure 7-7 I^2C Bit Transfer 7-17
Figure 7-8 I^2C Start and Stop Events 7-17
Figure 7-9 Acknowledgment on the I^2C Bus 7-18
Figure 7-10 I^2C Bus Protocol For Host Write Cycle 7-18
Figure 7-11 I^2C Bus Protocol For Host Read Cycle 7-19
Figure 8-1 ESAI Block Diagram 8-2
Figure 8-2 TCCR Register 8-8
Figure 8-3 ESAI Clock Generator Functional Block Diagram 8-9
Figure 8-4 ESAI Frame Sync Generator Functional Block Diagram 8-11
Figure 8-5 TCR Register 8-13
Figure 8-6 Normal and Network Operation 8-17
Figure 8-7 Frame Length Selection 8-20
Figure 8-8 RCCR Register 8-22
Figure 8-9 RCR Register 8-26
Figure 8-10 SAICR Register 8-32
Figure 8-11 SAICR SYN Bit Operation 8-34
Figure 8-12 SAISR Register 8-35
Figure 8-13 ESAI Data Path Programming Model ([R/T]SHFD=0) 8-38
Figure 8-14 ESAI Data Path Programming Model ([R/T]SHFD=1) 8-39
Figure 8-15 TSMA Register 8-41
Figure 8-16 TSMB Register 8-41
Figure 8-17 RSMA Register 8-42
Figure 8-18 RSMB Register 8-42
Figure 8-19 PCRC Register 8-48
Figure 8-20 PRRC Register 8-48
Figure 8-21 PDRC Register 8-49
Figure 9-1 ESAI_1 Block Diagram 9-2
Figure 9-2 EMUXR Register 9-5
Figure 9-3 TCCR_1 Register 9-6
Figure 9-4 ESAI_1 Clock Generator Functional Block Diagram 9-7
Figure 9-5 ESAI_1 Frame Sync Generator Functional Block Diagram 9-8
Figure 9-6 TCR_1 Register 9-8
Figure 9-7 RCCR_1 Register 9-9
Figure 9-8 RCR_1 Register 9-10
Figure 9-9 SAICR_1 Register 9-10
Figure 9-10 SAISR_1 Register 9-11

Figure 9-11 TSMA_1 Register 9-12
Figure 9-12 TSMB_1 Register 9-12
Figure 9-13 RSMA_1 Register 9-13
Figure 9-14 RSMB_1 Register 9-13
Figure 9-15 PCRE Register 9-14
Figure 9-16 PRRE Register 9-14
Figure 9-17 PDRE Register 9-15
Figure 10-1 Digital Audio Transmitter (DAX) Block Diagram 10-2
Figure 10-2 DAX Programming Model 10-4
Figure 10-3 DAX Relative Timing 10-8
Figure 10-4 Preamble sequence 10-9
Figure 10-5 Clock Multiplexer Diagram 10-10
Figure 10-6 Examples of data organization in memory 10-12
Figure 10-7 Port D Direction Register (PRRD) 10-13
Figure 10-8 Port D Data Register (PDRD) 10-14
Figure 11-1 Timer/Event Counter Block Diagram 11-2
Figure 11-2 Timer Block Diagram 11-3
Figure 11-3 Timer Module Programmer's Model 11-4
Figure 11-4 Timer Prescaler Load Register (TPLR) 11-5
Figure 11-5 Time Prescaler Count Register (TPCR) 11-6
Figure D-1 Status Register (SR) D-16
Figure D-2 Operating Mode Register (OMR) D-17
Figure D-3 Interrupt Priority Register–Core (IPR–C) D-18
Figure D-4 Interrupt Priority Register – Peripherals (IPR–P) D-19
Figure D-5 Phase Lock Loop Control Register (PCTL) D-20
Figure D-6 Host Receive and Host Transmit Data Registers D-21
Figure D-7 Host Control and Status Registers D-22
Figure D-8 Host Base Address and Host Port Control D-23
Figure D-9 Host Interrupt Control and Interrupt Status D-24
Figure D-10 Host Interrupt Vector and Command Vector D-25
Figure D-11 Host Receive and Transmit Byte Registers D-26
Figure D-12 SHI Slave Address and Clock Control Registers D-27
Figure D-13 SHI Transmit and Receive Data Registers D-28
Figure D-14 SHI Host Control/Status Register D-29
Figure D-15 ESAI Transmit Clock Control Register D-30
Figure D-16 ESAI Transmit Control Register D-31
Figure D-17 ESAI Receive Clock Control Register D-32
Figure D-18 ESAI Receive Control Register D-33
Figure D-19 ESAI Common Control Register D-34

Figure D-20 ESAI Status Register D-35
Figure D-21 ESAI_1 Multiplex Control Register D-36
Figure D-22 ESAI_1 Transmit Clock Control Register D-37
Figure D-23 ESAI_1 Transmit Control Register D-38
Figure D-24 ESAI_1 Receive Clock Control Register D-39
Figure D-25 ESAI_1 Receive Control Register D-40
Figure D-26 ESAI_1 Common Control Register D-41
Figure D-27 ESAI_1 Status Register D-42
Figure D-28 DAX Non-Audio Data Register D-43
Figure D-29 DAX Control and Status Registers D-44
Figure D-30 Timer Prescaler Load and Prescaler Count Registers (TPLR, TPCR) D-45
Figure D-31 Timer Control/Status Register D-46
Figure D-32 Timer Load, Compare and Count Registers D-47
Figure D-33 GPIO Port B D-48
Figure D-34 GPIO Port C D-49
Figure D-35 GPIO Port D D-50
Figure D-36 GPIO Port E D-51

List of Tables

Table 2-1 DSP56364 Functional Signal Groupings 2-1
Table 2-2 Power Inputs 2-3
Table 2-3 Grounds 2-3
Table 2-4 Clock and PLL Signals 2-4
Table 2-5 External Address Bus Signals 2-5
Table 2-6 External Data Bus Signals 2-5
Table 2-7 External Bus Control Signals 2-5
Table 2-8 Interrupt and Mode Control 2-8
Table 2-9 Host Interface 2-9
Table 2-10 Serial Host Interface Signals 2-12
Table 2-11 Enhanced Serial Audio Interface Signals 2-15
Table 2-12 Enhanced Serial Audio Interface_1 Signals 2-19
Table 2-13 Digital Audio Interface (DAX) Signals 2-21
Table 2-14 Timer Signal 2-22
Table 2-15 JTAG/OnCE Interface 2-22
Table 3-1 Internal Memory Configurations 3-1
Table 3-2 On-chip RAM Memory Locations 3-2
Table 3-3 On-chip ROM Memory Locations 3-2
Table 3-4 Internal I/O Memory Map 3-12
Table 4-1 Operating Mode Register (OMR) 4-2
Table 4-2 DSP56366 Operating Modes 4-4
Table 4-3 DSP56366 Mode Descriptions 4-5
Table 4-4 Interrupt Priority Level Bits 4-6
Table 4-5 Interrupt Sources Priorities Within an IPL 4-8
Table 4-6 DSP56366 Interrupt Vectors 4-10
Table 4-7 DMA Request Sources 4-12
Table 4-8 Identification Register Configuration 4-13
Table 4-9 JTAG Identification Register Configuration 4-14
Table 4-10 DSP56366 BSR Bit Definition 4-14
Table 6-1 HDI08 Signal Summary 6-3
Table 6-2 Strobe Signals Support signals 6-4
Table 6-3 Host request support signals 6-4
Table 6-4 HDI08 IRQ 6-8
Table 6-5 HDM[2:0] Functionality 6-8
Table 6-6 HDR and HDDR Functionality 6-16
Table 6-7 DSP-Side Registers after Reset 6-17
Table 6-8 HDI08 Host Side Register Map 6-19
Table 6-9 TREQ RREQ Interrupt Mode (HDM[2:0]=000 or HM[1:0]=00) 6-21
Table 6-10 TREQ RREQ DMA Mode (HM1¼0 or HM0¼0) 6-21
Table 6-11 HDRQ 6-21

Table 6-12 Host Mode Bit Definition 6-22
Table 6-13 INIT Command Effect 6-23
Table 6-14 Host Request Status (HREQ) 6-25
Table 6-15 Host Side Registers After Reset 6-27
Table 7-1 SHI Interrupt Vectors 7-5
Table 7-2 SHI Internal Interrupt Priorities 7-5
Table 7-3 SHI Noise Reduction Filter Mode 7-10
Table 7-4 SHI Data Size 7-11
Table 7-5 HREQ Function In SHI Slave Modes 7-12
Table 7-6 HCSR Receive Interrupt Enable Bits 7-14
Table 8-1 Receiver Clock Sources (asynchronous mode only) 8-5
Table 8-2 Transmitter Clock Sources 8-6
Table 8-3 Transmitter High Frequency Clock Divider 8-11
Table 8-4 Transmit Network Mode Selection 8-16
Table 8-5 ESAI Transmit Slot and Word Length Selection 8-18
Table 8-6 Receiver High Frequency Clock Divider 8-24
Table 8-7 SCKR Pin Definition Table 8-25
Table 8-8 FSR Pin Definition Table 8-25
Table 8-9 HCKR Pin Definition Table 8-26
Table 8-10 ESAI Receive Network Mode Selection 8-28
Table 8-11 ESAI Receive Slot and Word Length Selection 8-29
Table 8-12 PCRC and PRRC Bits Functionality 8-48
Table 9-1 EMUXR ESA/ESAI_1 Pin Selection 9-5
Table 9-2 Transmitter Clock Sources 9-6
Table 9-3 Receiver Clock Sources (asynchronous mode only) 9-9
Table 9-4 PCRE and PRRE Bits Functionality 9-14
Table 10-1 DAX Interrupt Vectors 10-4
Table 10-2 DAX Interrupt Priority 10-4
Table 10-3 Clock Source Selection 10-7
Table 10-4 Preamble Bit Patterns 10-9
Table 10-5 Examples of DMA configuration 10-11
Table 10-6 DAX Port GPIO Control Register Functionality 10-13
Table 11-1 Prescaler Source Selection 11-6
Table 11-2 Timer Control Bits for Timer 0 11-8
Table 11-3 Timer Control Bits for Timers 1 and 2 11-8
Table 11-4 Inverter (INV) Bit Operation 11-9
Table D-1 Internal I/O Memory Map D-2
Table D-2 DSP56366 Interrupt Vectors D-8
Table D-3 Interrupt Sources Priorities Within an IPL D-10
Table D-4 HDI08 Programming Model D-12

Preface

This manual describes the DSP56366 24-bit digital signal processor (DSP), its memory, operating modes, and peripheral modules. The DSP56366 is a member of the DSP56300 family of programmable CMOS DSPs. Changes in core functionality specific to the DSP56366 are also described in this manual.

The DSP56366 is targeted to applications that require digital audio compression and decompression, sound field processing, acoustic equalization, and other digital audio algorithms.

This manual is intended to be used with the following publications:

- The *DSP56300 Family Manual (DSP56300FM),* which describes the CPU, core programming models, and instruction set details.
- The *DSP56366 Technical Data Sheet (DSP56366)*, which provides electrical specifications, timing, pinout, and packaging descriptions of the DSP56366.

These documents, as well as Freescale's DSP development tools, can be obtained through a local Freescale Semiconductor Sales Office or authorized distributor.

To receive the latest information on this DSP, access the Freescale DSP home page at the address given on the back cover of this document.

This manual contains the following sections and appendices.

SECTION 1—DSP56366 OVERVIEW

- Provides a brief description of the DSP56366, including a features list and block diagram. Lists related documentation needed to use this chip and describes the organization of this manual.

SECTION 2—SIGNAL/CONNECTION DESCRIPTIONS

- Describes the signals on the DSP56366 pins and how these signals are grouped into interfaces.

SECTION 3—MEMORY CONFIGURATION

- Describes the DSP56366 memory spaces, RAM and ROM configuration, memory configurations and their bit settings, and memory maps.

SECTION 4—CORE CONFIGURATION

- Describes the registers used to configure the DSP56300 core when programming the DSP56366, in particular the interrupt vector locations and the operation of the interrupt priority registers. Explains the operating modes and how they affect the processor's program and data memories.

SECTION 5—GENERAL PURPOSE INPUT/OUTPUT (GPIO)

- Describes the DSP56366 GPIO capability and the programming model for the GPIO signals (operation, registers, and control).

SECTION 6— HOST INTERFACE (HDI08)

- Describes the HDI08 parallel host interface.

SECTION 7—SERIAL HOST INTERFACE (SHI)

- Describes the serial input/output interface providing a path for communication and program/coefficient data transfers between the DSP and an external host processor. The SHI can also communicate with other serial peripheral devices.

SECTION 8—ENHANCED SERIAL AUDIO INTERFACE (ESAI)

- Describes one of the full-duplex serial port for serial communication with a variety of serial devices.

SECTION 9—ENHANCED SERIAL AUDIO INTERFACE 1 (ESAI_1)

- Describes the second full-duplex serial port for serial communication with a variety of serial devices.

SECTION 10—DIGITAL AUDIO TRANSMITTER (DAX)

- Describes the full-duplex serial port for serial communication with a variety of serial devices.

SECTION 11—TRIPLE TIMER MODULE (TEC)

APPENDIX A—BOOTSTRAP PROGRAM

- Lists the bootstrap code used for the DSP56366.

APPENDIX B—EQUATES

- Lists equates for the DSP56366.

APPENDIX C—JTAG/BSDL LISTING

- Provides the BSDL listing for the DSP56366.

APPENDIX D—PROGRAMMING REFERENCE

- Lists peripheral addresses, interrupt addresses, and interrupt priorities for the DSP56366. Contains programming sheets listing the contents of the major DSP56366 registers for programmer reference.

Manual Conventions

The following conventions are used in this manual:

- Bits within registers are always listed from most significant bit (MSB) to least significant bit (LSB).
- When several related bits are discussed, they are referenced as AA[n:m], where n>m. For purposes of description, the bits are presented as if they are contiguous within a register. However, this is not always the case. Refer to the programming model diagrams or to the programmer's sheets to see the exact location of bits within a register.
- When a bit is described as "set", its value is 1. When a bit is described as "cleared", its value is 0.

- The word “assert” means that a high true (active high) signal is pulled high to V_{CC} or that a low true (active low) signal is pulled low to ground. The word “deassert” means that a high true signal is pulled low to ground or that a low true signal is pulled high to V_{CC}.

High True/Low True Signal Conventions

Signal/Symbol	Logic State	Signal State	Voltage
$\overline{\text{PIN}}$[1]	True	Asserted	Ground[2]
PIN	False	Deasserted	V_{CC}[3]
PIN	True	Asserted	V_{CC}
PIN	False	Deasserted	Ground

1 PIN is a generic term for any pin on the chip.

2 Ground is an acceptable low voltage level. See the appropriate data sheet for the range of acceptable low voltage levels (typically a TTL logic low).

3 V_{CC} is an acceptable high voltage level. See the appropriate data sheet for the range of acceptable high voltage levels (typically a TTL logic high).

- Pins or signals that are asserted low (made active when pulled to ground)
  - In text, have an overbar (e.g., $\overline{\text{RESET}}$ is asserted low).
  - In code examples, have a tilde in front of their names. In example below, line 3 refers to the $\overline{\text{SS0}}$ pin (shown as `~SS0`).
- Sets of pins or signals are indicated by the first and last pins or signals in the set (e.g., HA1–HA8).
- Code examples are displayed in a monospaced font, as shown below:

Example Sample Code Listing

```
BFSET    #$0007,X:PCC Configure:                    line 1
         ; MISO0, MOSI0, SCK0 for SPI master        line 2
         ; ~SS0 as PC3 for GPIO                     line 3
```

- Hex values are indicated with a dollar sign ($) preceding the hex value, as follows: $FFFFFF is the X memory address for the core interrupt priority register (IPR-C).
- The word “reset” is used in four different contexts in this manual:
  - the reset signal, written as “$\overline{\text{RESET}}$,”
  - the reset instruction, written as “RESET,”
  - the reset operating state, written as “Reset,” and
  - the reset function, written as “reset.”

NOTES

1 DSP56366 Overview

1.1 Introduction

This manual describes the DSP56366 24-bit digital signal processor (DSP), its memory, operating modes, and peripheral modules. The DSP56366 is a member of the DSP56300 family of programmable CMOS DSPs. The DSP56366 is targeted to applications that require digital audio compression/decompression, sound field processing, acoustic equalization and other digital audio algorithms. Changes in core functionality specific to the DSP56366 are also described in this manual. See Figure 1-1 for the block diagram of the DSP56366.

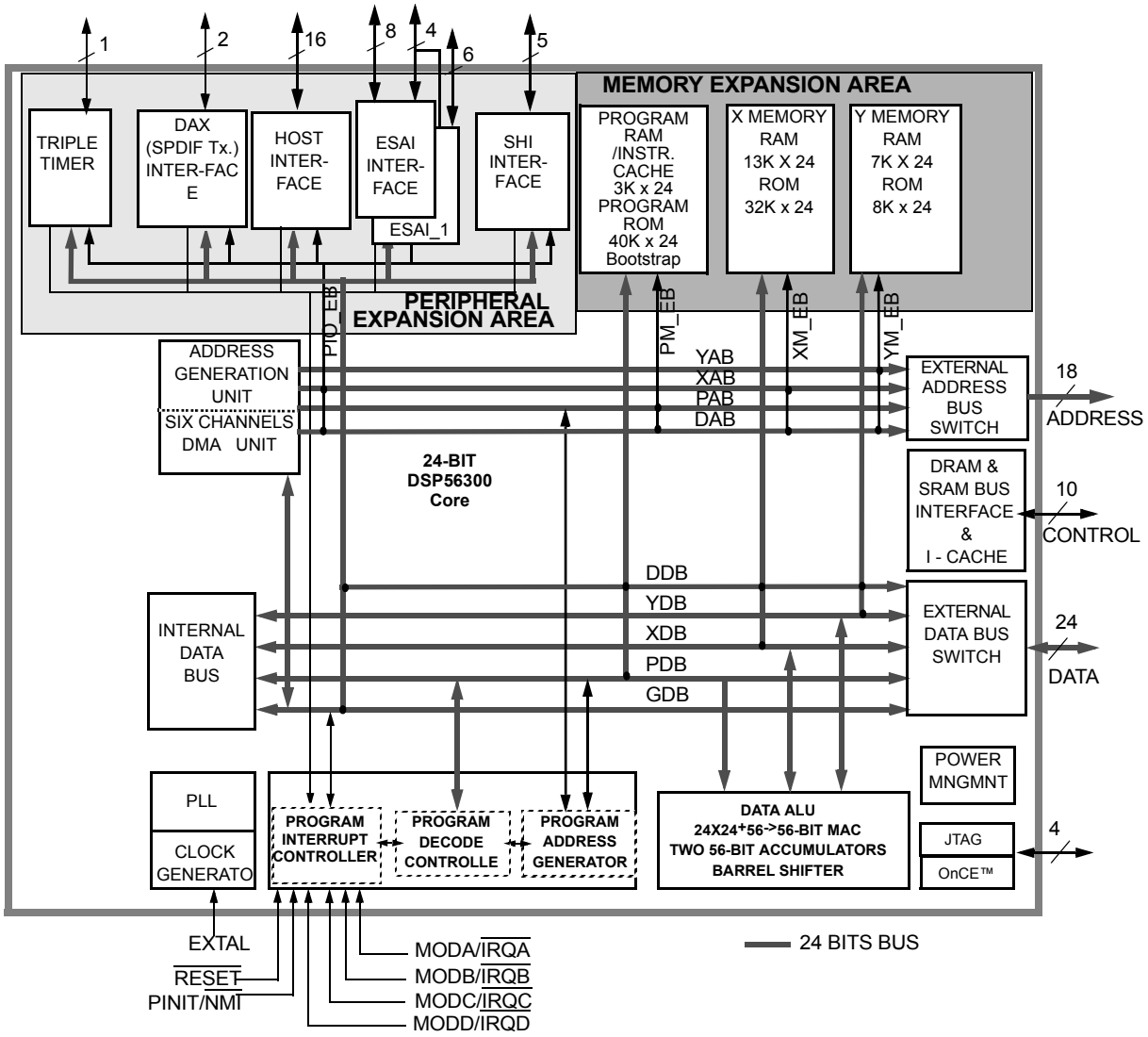


Figure 1-1 DSP56366 Block Diagram

1.2 DSP56300 Core Description

The DSP56366 uses the DSP56300 core, a high-performance, single clock cycle per instruction engine that provides up to twice the performance of Freescale's popular DSP56000 core family while retaining code compatibility with it.

The DSP56300 core family offers a new level of performance in speed and power, provided by its rich instruction set and low power dissipation, thus enabling a new generation of wireless, telecommunications, and multimedia products. For a description of the DSP56300 core, see Section 1.4, "DSP56300 Core Functional Blocks". Significant architectural enhancements to the DSP56300 core family include a barrel shifter, 24-bit addressing, an instruction cache, and direct memory access (DMA).

The DSP56300 core family members contain the DSP56300 core and additional modules. The modules are chosen from a library of standard predesigned elements such as memories and peripherals. New modules may be added to the library to meet customer specifications. A standard interface between the DSP56300 core and the on-chip memory and peripherals supports a wide variety of memory and peripheral configurations. Refer to Section 3, "Memory Configuration".

Core features are described fully in the *DSP56300 Family Manual*. Pinout, memory, and peripheral features are described in this manual.

- DSP56300 modular chassis
  - 120 Million Instructions Per Second (MIPS) with an 120 MHz clock at 3.3V.
  - Object Code Compatible with the 56K core.
  - Data ALU with a 24 × 24 bit multiplier-accumulator and a 56-bit barrel shifter. 16-bit arithmetic support.
  - Program Control with position independent code support and instruction cache support.
  - Six-channel DMA controller.
  - PLL based clocking with a wide range of frequency multiplications (1 to 4096), predivider factors (1 to 16) and power saving clock divider (2^i: i=0 to 7). Reduces clock noise.
  - Internal address tracing support and OnCE™ for Hardware/Software debugging.
  - JTAG port.
  - Very low-power CMOS design, fully static design with operating frequencies down to DC.
  - STOP and WAIT low-power standby modes.
- On-chip Memory Configuration
  - 7K × 24 Bit Y-Data RAM and 8K × 24 Bit Y-Data ROM.
  - 13K × 24 Bit X-Data RAM and 32K × 24 Bit X-Data ROM.
  - 40K × 24 Bit Program ROM.
  - 3K × 24 Bit Program RAM and 192 × 24 Bit Bootstrap ROM. 1K of Program RAM may be used as Instruction Cache or for Program ROM patching.
  - 2K × 24 Bit from Y Data RAM and 5K × 24 Bit from X Data RAM can be switched to Program RAM resulting in up to 10K × 24 Bit of Program RAM.
- Off-chip memory expansion
  - External Memory Expansion Port.

  - Off-chip expansion up to two 16M × 24-bit word of Data memory.
  - Off-chip expansion up to 16M × 24-bit word of Program memory.
  - Simultaneous glueless interface to SRAM and DRAM.
- Peripheral modules
  - Serial Audio Interface (ESAI): up to 4 receivers and up to 6 transmitters, master or slave. I^2S, Sony, AC97, network and other programmable protocols.
  - Serial Audio Interface I (ESAI_1): up to 4 receivers and up to 6 transmitters, master or slave. I^2S, Sony, AC97, network and other programmable protocols
    The ESAI_1 shares four of the data pins with ESAI, and ESAI_1 does NOT support HCKR and HCKT (high frequency clocks)
  - Serial Host Interface (SHI): SPI and I^2C protocols, multi master capability, 10-word receive FIFO, support for 8, 16 and 24-bit words.
  - Byte-wide parallel Host Interface (HDI08) with DMA support.
  - Triple Timer module (TEC).
  - Digital Audio Transmitter (DAX): 1 serial transmitter capable of supporting the SPDIF, IEC958, CP-340 and AES/EBU digital audio formats.
  - Pins of unused peripherals (except SHI) may be programmed as GPIO lines.
- 144-pin plastic TQFP package.

1.3 DSP56366 Audio Processor Architecture

This section defines the DSP56366 audio processor architecture. The audio processor is composed of the following units:

- The DSP56300 core is composed of the Data ALU, Address Generation Unit, Program Controller, Instruction-Cache Controller, DMA Controller, PLL-based clock oscillator, Memory Module Interface, Peripheral Module Interface and the On-Chip Emulator (OnCE). The DSP56300 core is described in the document *DSP56300 24-Bit Digital Signal Processor Family Manual*, Freescale publication DSP56300FM.
- Memory modules.
- Peripheral modules. The peripheral modules are defined in the following sections.

Memory sizes in the block diagram are defaults. Memory may be differently partitioned, according to the memory mode of the chip. See Section 1.4.8, "On-Chip Memory" for more details about memory size.

1.4 DSP56300 Core Functional Blocks

The DSP56300 core provides the following functional blocks:

- Data arithmetic logic unit (Data ALU)
- Address generation unit (AGU)
- Program control unit (PCU)
- Bus interface unit (BIU)
- DMA controller (with six channels)

- Instruction cache controller
- PLL-based clock oscillator
- OnCE module
- JTAG TAP
- Memory

In addition, the DSP56366 provides a set of on-chip peripherals, described in Section 1.5, "Peripheral Overview".

1.4.1 Data ALU

The Data ALU performs all the arithmetic and logical operations on data operands in the DSP56300 core. The components of the Data ALU are as follows:

- Fully pipelined 24-bit × 24-bit parallel multiplier-accumulator (MAC)
- Bit field unit, comprising a 56-bit parallel barrel shifter (fast shift and normalization; bit stream generation and parsing)
- Conditional ALU instructions
- 24-bit or 16-bit arithmetic support under software control
- Four 24-bit input general purpose registers: X1, X0, Y1, and Y0
- Six Data ALU registers (A2, A1, A0, B2, B1, and B0) that are concatenated into two general purpose, 56-bit accumulators (A and B), accumulator shifters
- Two data bus shifter/limiter circuits

1.4.1.1 Data ALU Registers

The Data ALU registers can be read or written over the X memory data bus (XDB) and the Y memory data bus (YDB) as 24- or 48-bit operands (or as 16- or 32-bit operands in 16-bit arithmetic mode). The source operands for the Data ALU, which can be 24, 48, or 56 bits (16, 32, or 40 bits in 16-bit arithmetic mode), always originate from Data ALU registers. The results of all Data ALU operations are stored in an accumulator.

All the Data ALU operations are performed in two clock cycles in pipeline fashion so that a new instruction can be initiated in every clock, yielding an effective execution rate of one instruction per clock cycle. The destination of every arithmetic operation can be used as a source operand for the immediately following arithmetic operation without a time penalty (i.e., without a pipeline stall).

1.4.1.2 Multiplier-Accumulator (MAC)

The MAC unit comprises the main arithmetic processing unit of the DSP56300 core and performs all of the calculations on data operands. In the case of arithmetic instructions, the unit accepts as many as three input operands and outputs one 56-bit result of the following form- Extension:Most Significant Product:Least Significant Product (EXT:MSP:LSP).

The multiplier executes 24-bit × 24-bit, parallel, fractional multiplies, between two's-complement signed, unsigned, or mixed operands. The 48-bit product is right-justified and added to the 56-bit contents of either

the A or B accumulator. A 56-bit result can be stored as a 24-bit operand. The LSP can either be truncated or rounded into the MSP. Rounding is performed if specified.

1.4.2 Address Generation Unit (AGU)

The AGU performs the effective address calculations using integer arithmetic necessary to address data operands in memory and contains the registers used to generate the addresses. It implements four types of arithmetic: linear, modulo, multiple wrap-around modulo, and reverse-carry. The AGU operates in parallel with other chip resources to minimize address-generation overhead.

The AGU is divided into two halves, each with its own Address ALU. Each Address ALU has four sets of register triplets, and each register triplet is composed of an address register, an offset register, and a modifier register. The two Address ALUs are identical. Each contains a 24-bit full adder (called an offset adder).

A second full adder (called a modulo adder) adds the summed result of the first full adder to a modulo value that is stored in its respective modifier register. A third full adder (called a reverse-carry adder) is also provided.

The offset adder and the reverse-carry adder are in parallel and share common inputs. The only difference between them is that the carry propagates in opposite directions. Test logic determines which of the three summed results of the full adders is output.

Each Address ALU can update one address register from its respective address register file during one instruction cycle. The contents of the associated modifier register specifies the type of arithmetic to be used in the address register update calculation. The modifier value is decoded in the Address ALU.

1.4.3 Program Control Unit (PCU)

The PCU performs instruction prefetch, instruction decoding, hardware DO loop control, and exception processing. The PCU implements a seven-stage pipeline and controls the different processing states of the DSP56300 core. The PCU consists of the following three hardware blocks:

- Program decode controller (PDC)
- Program address generator (PAG)
- Program interrupt controller (PIC)

The PDC decodes the 24-bit instruction loaded into the instruction latch and generates all signals necessary for pipeline control. The PAG contains all the hardware needed for program address generation, system stack, and loop control. The PIC arbitrates among all interrupt requests (internal interrupts, as well as the five external requests: $\overline{\text{IRQA}}$, $\overline{\text{IRQB}}$, $\overline{\text{IRQC}}$, $\overline{\text{IRQD}}$, and $\overline{\text{NMI}}$), and generates the appropriate interrupt vector address.

PCU features include the following:

- Position independent code support
- Addressing modes optimized for DSP applications (including immediate offsets)
- On-chip instruction cache controller
- On-chip memory-expandable hardware stack

- Nested hardware DO loops
- Fast auto-return interrupts

The PCU implements its functions using the following registers:

- PC—program counter register
- SR—Status register
- LA—loop address register
- LC—loop counter register
- VBA—vector base address register
- SZ—stack size register
- SP—stack pointer
- OMR—operating mode register
- SC—stack counter register

The PCU also includes a hardware system stack (SS).

1.4.4 Internal Buses

To provide data exchange between blocks, the following buses are implemented:

- Peripheral input/output expansion bus (PIO_EB) to peripherals
- Program memory expansion bus (PM_EB) to program memory
- X memory expansion bus (XM_EB) to X memory
- Y memory expansion bus (YM_EB) to Y memory
- Global data bus (GDB) between registers in the DMA, AGU, OnCE, PLL, BIU, and PCU as well as the memory-mapped registers in the peripherals
- DMA data bus (DDB) for carrying DMA data between memories and/or peripherals
- DMA address bus (DAB) for carrying DMA addresses to memories and peripherals
- Program Data Bus (PDB) for carrying program data throughout the core
- X memory Data Bus (XDB) for carrying X data throughout the core
- Y memory Data Bus (YDB) for carrying Y data throughout the core
- Program address bus (PAB) for carrying program memory addresses throughout the core
- X memory address bus (XAB) for carrying X memory addresses throughout the core
- Y memory address bus (YAB) for carrying Y memory addresses throughout the core

All internal buses on the DSP56300 family members are 24-bit buses. See Figure 1-1.

1.4.5 Direct Memory Access (DMA)

The DMA block has the following features:

- Six DMA channels supporting internal and external accesses
- One-, two-, and three-dimensional transfers (including circular buffering)

- End-of-block-transfer interrupts
- Triggering from interrupt lines and all peripherals

1.4.6 PLL-based Clock Oscillator

The clock generator in the DSP56300 core is composed of two main blocks: the PLL, which performs clock input division, frequency multiplication, and skew elimination; and the clock generator (CLKGEN), which performs low-power division and clock pulse generation. PLL-based clocking:

- Allows change of low-power divide factor (DF) without loss of lock
- Provides output clock with skew elimination
- Provides a wide range of frequency multiplications (1 to 4096), predivider factors (1 to 16), and a power-saving clock divider (2^i: i = 0 to 7) to reduce clock noise

The PLL allows the processor to operate at a high internal clock frequency using a low frequency clock input. This feature offers two immediate benefits:

- A lower frequency clock input reduces the overall electromagnetic interference generated by a system.
- The ability to oscillate at different frequencies reduces costs by eliminating the need to add additional oscillators to a system.

1.4.7 JTAG TAP and OnCE Module

The DSP56300 core provides a dedicated user-accessible TAP fully compatible with the *IEEE 1149.1 Standard Test Access Port and Boundary Scan Architecture*. Problems associated with testing high-density circuit boards led to developing this standard under the sponsorship of the Test Technology Committee of IEEE and JTAG. The DSP56300 core implementation supports circuit-board test strategies based on this standard.

The test logic includes a TAP consisting of four dedicated signals, a 16-state controller, and three test data registers. A boundary scan register links all device signals into a single shift register. The test logic, implemented utilizing static logic design, is independent of the device system logic. More information on the JTAG port is provided in *DSP56300 Family Manual, JTAG Port*.

The OnCE module provides a nonintrusive means of interacting with the DSP56300 core and its peripherals so a user can examine registers, memory, or on-chip peripherals. This facilitates hardware and software development on the DSP56300 core processor. OnCE module functions are provided through the JTAG TAP signals. More information on the OnCE module is provided in *DSP56300 Family Manual, On-Chip Emulation Module*.

1.4.8 On-Chip Memory

The memory space of the DSP56300 core is partitioned into program memory space,
X data memory space, and Y data memory space. The data memory space is divided into X and Y data memory in order to work with the two Address ALUs and to feed two operands simultaneously to the Data

ALU. Memory space includes internal RAM and ROM and can be expanded off-chip under software control.

There is an instruction cache, made using program RAM. The patch mode (which uses instruction cache space) is used to patch program ROM. The memory switch mode is used to increase the size of program RAM as needed (switch from X data RAM and/or Y data RAM).

There are on-chip ROMs for program memory (40K × 24-bit), bootstrap memory (192 words × 24-bit), X ROM (32K × 24-bit), and Y ROM(8K × 24-bit).

More information on the internal memory is provided in Section 3, "Memory Configuration".

1.4.9 Off-Chip Memory Expansion

Memory can be expanded off-chip as follows:

- Data memory can be expanded to two 16 M × 24-bit word memory spaces in 24-bit address mode (64K in 16-bit address mode).
- Program memory can be expanded to one 16 M × 24-bit word memory space in 24-bit address mode (64K in 16-bit address mode).

Other features of external memory expansion include the following:

- External memory expansion port
- Chip-select logic glueless interface to static random access memory (SRAM)
- On-chip dynamic RAM (DRAM) controller for glueless interface to DRAM
- Eighteen external address lines

1.5 Peripheral Overview

The DSP56366 is designed to perform a wide variety of fixed-point digital signal processing functions. In addition to the core features previously discussed, the DSP56366 provides the following peripherals:

- 8-bit parallel host interface (HDI08, with DMA support) to external hosts
- As many as 37 user-configurable general purpose input/output (GPIO) signals
- Timer/event counter (TEC) module, containing three independent timers
- Memory switch mode in on-chip memory
- Four external interrupt/mode control lines and one external non-maskable interrupt line
- Enhanced serial audio interface (ESAI) with up to four receivers and up to six transmitters, master or slave, using the I^2S, Sony, AC97, network, and other programmable protocols
- A second enhanced serial audio interface (ESAI_1) with 6 dedicated pins.
- Serial host interface (SHI) using SPI and I^2C protocols, with multi-master capability, 10-word receive FIFO, and support for 8-, 16-, and 24-bit words
- Digital audio transmitter (DAX): a serial transmitter capable of supporting the SPDIF, IEC958, CP-340, and AES/EBU digital audio formats

1.5.1 Host Interface (HDI08)

The host interface (HDI08) is a byte-wide, full-duplex, double-buffered, parallel port that can be connected directly to the data bus of a host processor. The HDI08 supports a variety of buses and provides glueless connection with a number of industry-standard DSPs, microcomputers, microprocessors, and DMA hardware.

The DSP core treats the HDI08 as a memory-mapped peripheral, using either standard polled or interrupt programming techniques. Separate transmit and receive data registers are double-buffered to allow the DSP and host processor to efficiently transfer data at high speed. Memory mapping allows DSP core communication with the HDI08 registers to be accomplished using standard instructions and addressing modes.

Since the host bus may operate asynchronously with the DSP core clock, the HDI08 registers are divided into 2 banks. The “host side” bank is accessible to the external host, and the “DSP side” bank is accessible to the DSP core.

The HDI08 supports the following three classes of interfaces:

- Host processor/MCU connection
- DMA controller
- GPIO port

Host port pins not in use may be configured as GPIO pins. The host interface provides up to 16 GPIO pins. These pins can be programmed to function as either GPIO or host interface.

For more information on the HDI08, see Section 6, Host Interface (HDI08)**.**

1.5.2 General Purpose Input/Output (GPIO)

The GPIO port consists of as many as 37 programmable signals, all of which are also used by the peripherals (HDI08, ESAI, ESAI_1, DAX, and TEC). There are no dedicated GPIO signals. The signals are configured as GPIO after hardware reset. Register programming techniques for all GPIO functionality among these interfaces are very similar.

1.5.3 Triple Timer (TEC)

This section describes a peripheral module composed of a common 21-bit prescaler and three independent and identical general purpose 24-bit timer/event counters, each one having its own register set.

Each timer can use internal or external clocking and can interrupt the DSP after a specified number of events (clocks). Timer 0 can signal an external device after counting internal events. Each timer can also be used to trigger DMA transfers after a specified number of events (clocks) occurred. One timer (Timer 0) connects to the external world through one bidirectional pin TIO0. When TIO0 is configured as input, the timer functions as an external event counter or can measure external pulse width/signal period. When TIO0 is used as output the timer is functioning as either a timer, a watchdog or a Pulse Width Modulator. When the TIO0 pin is not used by the timer it can be used as a General Purpose Input/Output Pin. Refer to Section 11, "Timer/ Event Counter".

1.5.4 Enhanced Serial Audio Interface (ESAI)

The ESAI provides a full-duplex serial port for serial communication with a variety of serial devices including one or more industry-standard codecs, other DSPs, microprocessors, and peripherals that implement the Freescale SPI serial protocol. The ESAI consists of independent transmitter and receiver sections, each with its own clock generator. It is a superset of the DSP56300 family ESSI peripheral and of the DSP56000 family SAI peripheral. For more information on the ESAI, refer to Section 8, "Enhanced Serial AUDIO Interface (ESAI)".

1.5.5 Enhanced Serial Audio Interface 1 (ESAI_1)

The ESAI_1 is a second ESAI interface with just 6 dedicated pins instead of the 12 pins of the full ESAI. Four data pins are shared with the ESAI, while the two high frequency clock pins are not available. Other than the available pins, ESAI_1 is functionally identical to ESAI. For more information on the ESAI_1, refer to Section 9, "Enhanced Serial Audio Interface 1 (ESAI_1)".

1.5.6 Serial Host Interface (SHI)

The SHI is a serial input/output interface providing a path for communication and program/coefficient data transfers between the DSP and an external host processor. The SHI can also communicate with other serial peripheral devices. The SHI can interface directly to either of two well-known and widely used synchronous serial buses: the Freescale serial peripheral interface (SPI) bus and the Philips inter-integrated-circuit control (I^2C) bus. The SHI supports either the SPI or I^2C bus protocol, as required, from a slave or a single-master device. To minimize DSP overhead, the SHI supports single-, double-, and triple-byte data transfers. The SHI has a 10-word receive FIFO that permits receiving up to 30 bytes before generating a receive interrupt, reducing the overhead for data reception. For more information on the SHI, refer to Section 7, "Serial Host Interface".

1.5.7 Digital Audio Transmitter (DAX)

The DAX is a serial audio interface module that outputs digital audio data in the AES/EBU, CP-340 and IEC958 formats. For more information on the DAX, refer to Section 10, "Digital Audio Transmitter".

2 Signal/Connection Descriptions

2.1 Signal Groupings

The input and output signals of the DSP56364 are organized into functional groups, which are listed in Table 2-1 and illustrated in Figure 2-1.

The DSP56364 is operated from a 3.3 V supply; however, some of the inputs can tolerate 5 V. A special notice for this feature is added to the signal descriptions of those inputs.

Table 2-1 DSP56364 Functional Signal Groupings

Functional Group		Number of Signals	Detailed Description
Power (V_{CC})		20	Table 2-2
Ground (GND)		18	Table 2-3
Clock and PLL		3	Table 2-4
Address bus	Port A[1]	18	Table 2-5
Data bus		24	Table 2-6
Bus control		10	Table 2-7
Interrupt and mode control		5	Table 2-8
HDI08	Port B[2]	16	Table 2-9
SHI		5	Table 2-10
ESAI	Port C[3]	12	Table 2-11
ESAI_1	Port E[4]	6	Table 2-12
Digital audio transmitter (DAX)	Port D[5]	2	Table 2-13
Timer		1	Table 2-14
JTAG/OnCE Port		4	Table 2-15

1 Port A is the external memory interface port, including the external address bus, data bus, and control signals.
2 Port B signals are the GPIO port signals which are multiplexed with the HDI08 signals.
3 Port C signals are the GPIO port signals which are multiplexed with the ESAI signals.
4 Port E signals are the GPIO port signals which are multiplexed with the ESAI_1 signals.
5 Port D signals are the GPIO port signals which are multiplexed with the DAX signals.

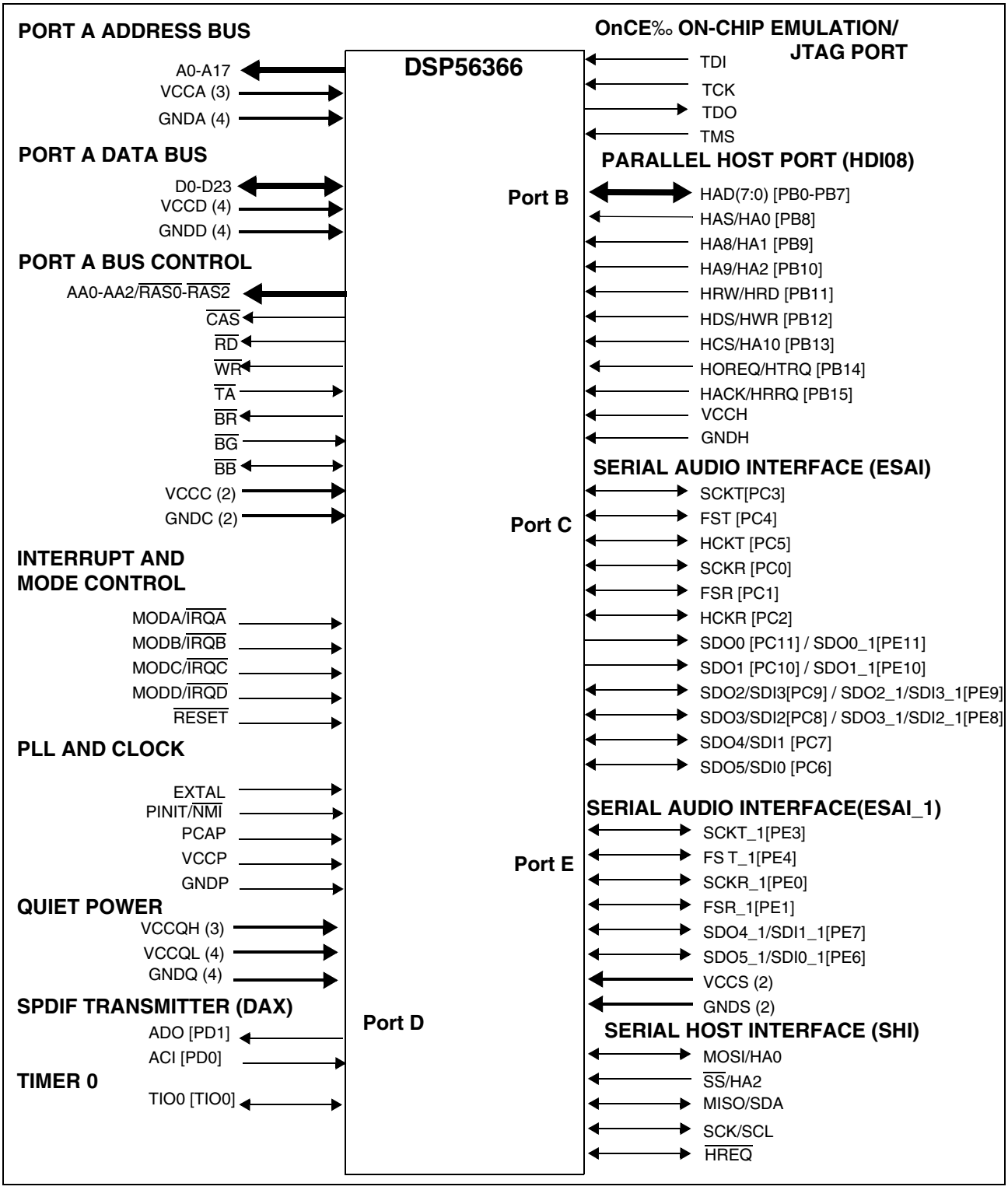


Figure 2-1 Signals Identified by Functional Group

2.2 Power

Table 2-2 Power Inputs

Power Name	Description
V_{CCP}	**PLL Power** — V_{CCP} is V_{CC} dedicated for PLL use. The voltage should be well-regulated and the input should be provided with an extremely low impedance path to the V_{CC} power rail. There is one V_{CCP} input.
V_{CCQL} (4)	**Quiet Core (Low) Power** — V_{CCQL} is an isolated power for the internal processing logic. This input must be tied externally to all other chip power inputs. The user must provide adequate external decoupling capacitors. There are four V_{CCQL} inputs.
V_{CCQH} (3)	**Quiet External (High) Power** — V_{CCQH} is a quiet power source for I/O lines. This input must be tied externally to all other chip power inputs. The user must provide adequate decoupling capacitors. There are three V_{CCQH} inputs.
V_{CCA} (3)	**Address Bus Power** — V_{CCA} is an isolated power for sections of the address bus I/O drivers. This input must be tied externally to all other chip power inputs. The user must provide adequate external decoupling capacitors. There are three V_{CCA} inputs.
V_{CCD} (4)	**Data Bus Power** — V_{CCD} is an isolated power for sections of the data bus I/O drivers. This input must be tied externally to all other chip power inputs. The user must provide adequate external decoupling capacitors. There are four V_{CCD} inputs.
V_{CCC} (2)	**Bus Control Power** — V_{CCC} is an isolated power for the bus control I/O drivers. This input must be tied externally to all other chip power inputs. The user must provide adequate external decoupling capacitors. There are two V_{CCC} inputs.
V_{CCH}	**Host Power** — V_{CCH} is an isolated power for the HDI08 I/O drivers. This input must be tied externally to all other chip power inputs. The user must provide adequate external decoupling capacitors. There is one V_{CCH} input.
V_{CCS} (2)	**SHI, ESAI, ESAI_1, DAX and Timer Power** — V_{CCS} is an isolated power for the SHI, ESAI, ESAI_1, DAX and Timer. This input must be tied externally to all other chip power inputs. The user must provide adequate external decoupling capacitors. There are two V_{CCS} inputs.

2.3 Ground

Table 2-3 Grounds

Ground Name	Description
GND_P	**PLL Ground** — GND_P is a ground dedicated for PLL use. The connection should be provided with an extremely low-impedance path to ground. V_{CCP} should be bypassed to GND_P by a 0.47 µF capacitor located as close as possible to the chip package. There is one GND_P connection.
GND_Q (4)	**Quiet Ground** — GND_Q is an isolated ground for the internal processing logic. This connection must be tied externally to all other chip ground connections. The user must provide adequate external decoupling capacitors. There are four GND_Q connections.

Table 2-3 Grounds (continued)

Ground Name	Description
GND_A (4)	**Address Bus Ground** — GND_A is an isolated ground for sections of the address bus I/O drivers. This connection must be tied externally to all other chip ground connections. The user must provide adequate external decoupling capacitors. There are four GND_A connections.
GND_D (4)	**Data Bus Ground** — GND_D is an isolated ground for sections of the data bus I/O drivers. This connection must be tied externally to all other chip ground connections. The user must provide adequate external decoupling capacitors. There are four GND_D connections.
GND_C (2)	**Bus Control Ground** — GND_C is an isolated ground for the bus control I/O drivers. This connection must be tied externally to all other chip ground connections. The user must provide adequate external decoupling capacitors. There are two GND_C connections.
GND_H	**Host Ground** — GND_h is an isolated ground for the HD08 I/O drivers. This connection must be tied externally to all other chip ground connections. The user must provide adequate external decoupling capacitors. There is one GND_H connection.
GND_S (2)	**SHI, ESAI, ESAI_1, DAX and Timer Ground** — GND_S is an isolated ground for the SHI, ESAI, ESAI_1, DAX and Timer. This connection must be tied externally to all other chip ground connections. The user must provide adequate external decoupling capacitors. There are two GND_S connections.

2.4 Clock and PLL

Table 2-4 Clock and PLL Signals

Signal Name	Type	State during Reset	Signal Description
EXTAL	Input	Input	**External Clock Input** — An external clock source must be connected to EXTAL in order to supply the clock to the internal clock generator and PLL. *This input cannot tolerate 5 V.*
PCAP	Input	Input	**PLL Capacitor** — PCAP is an input connecting an off-chip capacitor to the PLL filter. Connect one capacitor terminal to PCAP and the other terminal to V_{CCP} If the PLL is not used, PCAP may be tied to V_{CC}, GND, or left floating.
PINIT/$\overline{\text{NMI}}$	Input	Input	**PLL Initial/Nonmaskable Interrupt** — During assertion of $\overline{\text{RESET}}$, the value of PINIT/$\overline{\text{NMI}}$ is written into the PLL Enable (PEN) bit of the PLL control register, determining whether the PLL is enabled or disabled. After $\overline{\text{RESET}}$ de assertion and during normal instruction processing, the PINIT/$\overline{\text{NMI}}$ Schmitt-trigger input is a negative-edge-triggered nonmaskable interrupt (NMI) request internally synchronized to internal system clock. *This input cannot tolerate 5 V.*

2.5 External Memory Expansion Port (Port A)

When the DSP56364 enters a low-power standby mode (stop or wait), it releases bus mastership and tri-states the relevant port A signals: A0 – A17, D0 – D23, AA0/$\overline{\text{RAS0}}$ – AA2/$\overline{\text{RAS2}}$, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{BB}}$, $\overline{\text{CAS}}$.

2.5.1 External Address Bus

Table 2-5 External Address Bus Signals

Signal Name	Type	State during Reset	Signal Description
A0–A17	Output	Tri-stated	**Address Bus** — When the DSP is the bus master, A0 – A17 are active-high outputs that specify the address for external program and data memory accesses. Otherwise, the signals are tri-stated. To minimize power dissipation, A0 – A17 do not change state when external memory spaces are not being accessed.

2.5.2 External Data Bus

Table 2-6 External Data Bus Signals

Signal Name	Type	State during Reset	Signal Description
D0–D23	Input/Output	Tri-stated	**Data Bus** — When the DSP is the bus master, D0 – D23 are active-high, bidirectional input/outputs that provide the bidirectional data bus for external program and data memory accesses. Otherwise, D0 – D23 are tri-stated.

2.5.3 External Bus Control

Table 2-7 External Bus Control Signals

Signal Name	Type	State during Reset	Signal Description
AA0–AA2/$\overline{\text{RAS0}}$ – $\overline{\text{RAS2}}$	Output	Tri-stated	**Address Attribute or Row Address Strobe** — When defined as AA, these signals can be used as chip selects or additional address lines. When defined as $\overline{\text{RAS}}$, these signals can be used as $\overline{\text{RAS}}$ for DRAM interface. These signals are tri-statable outputs with programmable polarity.
$\overline{\text{CAS}}$	Output	Tri-stated	**Column Address Strobe** — When the DSP is the bus master, $\overline{\text{CAS}}$ is an active-low output used by DRAM to strobe the column address. Otherwise, if the bus mastership enable (BME) bit in the DRAM control register is cleared, the signal is tri-stated.
$\overline{\text{RD}}$	Output	Tri-stated	**Read Enable** — When the DSP is the bus master, $\overline{\text{RD}}$ is an active-low output that is asserted to read external memory on the data bus (D0-D23). Otherwise, $\overline{\text{RD}}$ is tri-stated.

Table 2-7 External Bus Control Signals (continued)

Signal Name	Type	State during Reset	Signal Description
$\overline{WR}$	Output	Tri-stated	**Write Enable** — When the DSP is the bus master, $\overline{WR}$ is an active-low output that is asserted to write external memory on the data bus (D0-D23). Otherwise, $\overline{WR}$ is tri-stated.
$\overline{TA}$	Input	Ignored Input	**Transfer Acknowledge** — If the DSP is the bus master and there is no external bus activity, or the DSP is not the bus master, the $\overline{TA}$ input is ignored. The $\overline{TA}$ input is a data transfer acknowledge (DTACK) function that can extend an external bus cycle indefinitely. Any number of wait states (1, 2. . .infinity) may be added to the wait states inserted by the BCR by keeping $\overline{TA}$ deasserted. In typical operation, $\overline{TA}$ is deasserted at the start of a bus cycle, is asserted to enable completion of the bus cycle, and is deasserted before the next bus cycle. The current bus cycle completes one clock period after $\overline{TA}$ is asserted synchronous to the internal system clock. The number of wait states is determined by the $\overline{TA}$ input or by the bus control register (BCR), whichever is longer. The BCR can be used to set the minimum number of wait states in external bus cycles. In order to use the $\overline{TA}$ functionality, the BCR must be programmed to at least one wait state. A zero wait state access cannot be extended by $\overline{TA}$ deassertion, otherwise improper operation may result. $\overline{TA}$ can operate synchronously or asynchronously, depending on the setting of the TAS bit in the operating mode register (OMR). $\overline{TA}$ functionality may not be used while performing DRAM type accesses, otherwise improper operation may result.
$\overline{BR}$	Output	Output (deasserted)	**Bus Request** — $\overline{BR}$ is an active-low output, never tri-stated. $\overline{BR}$ is asserted when the DSP requests bus mastership. $\overline{BR}$ is deasserted when the DSP no longer needs the bus. $\overline{BR}$ may be asserted or deasserted independent of whether the DSP56364 is a bus master or a bus slave. Bus “parking” allows $\overline{BR}$ to be deasserted even though the DSP56364 is the bus master. (See the description of bus “parking” in the $\overline{BB}$ signal description.) The bus request hold (BRH) bit in the BCR allows $\overline{BR}$ to be asserted under software control even though the DSP does not need the bus. $\overline{BR}$ is typically sent to an external bus arbitrator that controls the priority, parking, and tenure of each master on the same external bus. $\overline{BR}$ is only affected by DSP requests for the external bus, never for the internal bus. During hardware reset, $\overline{BR}$ is deasserted and the arbitration is reset to the bus slave state.

Table 2-7 External Bus Control Signals (continued)

Signal Name	Type	State during Reset	Signal Description
$\overline{BG}$	Input	Ignored Input	**Bus Grant** — $\overline{BG}$ is an active-low input. $\overline{BG}$ is asserted by an external bus arbitration circuit when the DSP56364 becomes the next bus master. When $\overline{BG}$ is asserted, the DSP56364 must wait until $\overline{BB}$ is deasserted before taking bus mastership. When $\overline{BG}$ is deasserted, bus mastership is typically given up at the end of the current bus cycle. This may occur in the middle of an instruction that requires more than one external bus cycle for execution. For proper $\overline{BG}$ operation, the asynchronous bus arbitration enable bit (ABE) in the OMR register must be set.
$\overline{BB}$	Input/ Output	Input	**Bus Busy** — $\overline{BB}$ is a bidirectional active-low input/output. $\overline{BB}$ indicates that the bus is active. Only after $\overline{BB}$ is deasserted can the pending bus master become the bus master (and then assert the signal again). The bus master may keep $\overline{BB}$ asserted after ceasing bus activity regardless of whether $\overline{BR}$ is asserted or deasserted. This is called “bus parking” and allows the current bus master to reuse the bus without rearbitration until another device requires the bus. The deassertion of $\overline{BB}$ is done by an “active pull-up” method (i.e., $\overline{BB}$ is driven high and then released and held high by an external pull-up resistor). For proper $\overline{BB}$ operation, the asynchronous bus arbitration enable bit (ABE) in the OMR register must be set. $\overline{BB}$ requires an external pull-up resistor.

2.6 Interrupt and Mode Control

The interrupt and mode control signals select the chip’s operating mode as it comes out of hardware reset. After $\overline{RESET}$ is deasserted, these inputs are hardware interrupt request lines.

Table 2-8 Interrupt and Mode Control

Signal Name	Type	State during Reset	Signal Description
MODA/$\overline{\text{IRQA}}$	Input	Input	**Mode Select A/External Interrupt Request A** — MODA/$\overline{\text{IRQA}}$ is an active-low Schmitt-trigger input, internally synchronized to the DSP clock. MODA/$\overline{\text{IRQA}}$ selects the initial chip operating mode during hardware reset and becomes a level-sensitive or negative-edge-triggered, maskable interrupt request input during normal instruction processing. MODA, MODB, MODC, and MODD select one of 16 initial chip operating modes, latched into the OMR when the $\overline{\text{RESET}}$ signal is deasserted. If the processor is in the stop standby state and the MODA/$\overline{\text{IRQA}}$ pin is pulled to GND, the processor will exit the stop state. *This input is 5 V tolerant.*
MODB/$\overline{\text{IRQB}}$	Input	Input	**Mode Select B/External Interrupt Request B** — MODB/$\overline{\text{IRQB}}$ is an active-low Schmitt-trigger input, internally synchronized to the DSP clock. MODB/$\overline{\text{IRQB}}$ selects the initial chip operating mode during hardware reset and becomes a level-sensitive or negative-edge-triggered, maskable interrupt request input during normal instruction processing. MODA, MODB, MODC, and MODD select one of 16 initial chip operating modes, latched into OMR when the $\overline{\text{RESET}}$ signal is deasserted. This input is 5 V tolerant.
MODC/$\overline{\text{IRQC}}$	Input	Input	**Mode Select C/External Interrupt Request C** — MODC/$\overline{\text{IRQC}}$ is an active-low Schmitt-trigger input, internally synchronized to the DSP clock. MODC/$\overline{\text{IRQC}}$ selects the initial chip operating mode during hardware reset and becomes a level-sensitive or negative-edge-triggered, maskable interrupt request input during normal instruction processing. MODA, MODB, MODC, and MODD select one of 16 initial chip operating modes, latched into OMR when the $\overline{\text{RESET}}$ signal is deasserted. This input is 5 V tolerant.
MODD/$\overline{\text{IRQD}}$	Input	Input	**Mode Select D/External Interrupt Request D** — MODD/$\overline{\text{IRQD}}$ is an active-low Schmitt-trigger input, internally synchronized to the DSP clock. MODD/$\overline{\text{IRQD}}$ selects the initial chip operating mode during hardware reset and becomes a level-sensitive or negative-edge-triggered, maskable interrupt request input during normal instruction processing. MODA, MODB, MODC, and MODD select one of 16 initial chip operating modes, latched into OMR when the $\overline{\text{RESET}}$ signal is deasserted. This input is 5 V tolerant.
$\overline{\text{RESET}}$	Input	Input	**Reset** — $\overline{\text{RESET}}$ is an active-low, Schmitt-trigger input. When asserted, the chip is placed in the Reset state and the internal phase generator is reset. The Schmitt-trigger input allows a slowly rising input (such as a capacitor charging) to reset the chip reliably. When the $\overline{\text{RESET}}$ signal is deasserted, the initial chip operating mode is latched from the MODA, MODB, MODC, and MODD inputs. The $\overline{\text{RESET}}$ signal must be asserted during power up. A stable EXTAL signal must be supplied while $\overline{\text{RESET}}$ is being asserted. *This input is 5 V tolerant.*

2.7 PARALLEL HOST INTERFACE (HDI08)

The HDI08 provides a fast, 8-bit, parallel data port that may be connected directly to the host bus. The HDI08 supports a variety of standard buses and can be directly connected to a number of industry standard microcomputers, microprocessors, DSPs, and DMA hardware.

Table 2-9 Host Interface

Signal Name	Type	State during Reset	Signal Description
H0 – H7	Input/ output		**Host Data** — When HDI08 is programmed to interface a nonmultiplexed host bus and the HI function is selected, these signals are lines 0 – 7 of the bidirectional, tri-state data bus.
HAD0 – HAD7	Input/ output		**Host Address/Data** — When HDI08 is programmed to interface a multiplexed host bus and the HI function is selected, these signals are lines 0 – 7 of the address/data bidirectional, multiplexed, tri-state bus.
PB0 – PB7	Input, output, or disconnected	GPIO disconnected	**Port B 0–7** — When the HDI08 is configured as GPIO, these signals are individually programmable as input, output, or internally disconnected.
			The default state after reset for these signals is GPIO disconnected. These inputs are 5 V tolerant.
HA0	Input	GPIO disconnected	**Host Address Input 0** — When the HDI08 is programmed to interface a nonmultiplexed host bus and the HI function is selected, this signal is line 0 of the host address input bus.
$\overline{\text{HAS}}$/HAS	Input		**Host Address Strobe** — When HDI08 is programmed to interface a multiplexed host bus and the HI function is selected, this signal is the host address strobe (HAS) Schmitt-trigger input. The polarity of the address strobe is programmable, but is configured active-low ($\overline{\text{HAS}}$) following reset.
PB8	Input, output, or disconnected		**Port B 8** — When the HDI08 is configured as GPIO, this signal is individually programmed as input, output, or internally disconnected.
			The default state after reset for this signal is GPIO disconnected. *This input is 5 V tolerant.*
HA1	Input	GPIO disconnected	**Host Address Input 1** — When the HDI08 is programmed to interface a nonmultiplexed host bus and the HI function is selected, this signal is line 1 of the host address (HA1) input bus.
HA8	Input		**Host Address 8** — When HDI08 is programmed to interface a multiplexed host bus and the HI function is selected, this signal is line 8 of the host address (HA8) input bus.
PB9	Input, output, or disconnected		**Port B 9** — When the HDI08 is configured as GPIO, this signal is individually programmed as input, output, or internally disconnected.
			The default state after reset for this signal is GPIO disconnected. This input is 5 V tolerant.

Table 2-9 Host Interface (continued)

Signal Name	Type	State during Reset	Signal Description
HA2	Input	GPIO disconnected	**Host Address Input 2** — When the HDI08 is programmed to interface a non-multiplexed host bus and the HI function is selected, this signal is line 2 of the host address (HA2) input bus.
HA9	Input		**Host Address 9** — When HDI08 is programmed to interface a multiplexed host bus and the HI function is selected, this signal is line 9 of the host address (HA9) input bus.
PB10	Input, Output, or Disconnected		**Port B 10** — When the HDI08 is configured as GPIO, this signal is individually programmed as input, output, or internally disconnected.
			The default state after reset for this signal is GPIO disconnected. *This input is 5 V tolerant.*
HRW	Input	GPIO disconnected	**Host Read/Write** — When HDI08 is programmed to interface a single-data-strobe host bus and the HI function is selected, this signal is the Host Read/$\overline{\text{Write}}$ (HRW) input.
$\overline{\text{HRD}}$/ HRD	Input		**Host Read Data** — When HDI08 is programmed to interface a double-data-strobe host bus and the HI function is selected, this signal is the host read data strobe (HRD) Schmitt-trigger input. The polarity of the data strobe is programmable, but is configured as active-low ($\overline{\text{HRD}}$) after reset.
PB11	Input, Output, or Disconnected		**Port B 11** — When the HDI08 is configured as GPIO, this signal is individually programmed as input, output, or internally disconnected.
			The default state after reset for this signal is GPIO disconnected. This input is 5 V tolerant.
$\overline{\text{HDS}}$/ HDS	Input	GPIO disconnected	**Host Data Strobe** — When HDI08 is programmed to interface a single-data-strobe host bus and the HI function is selected, this signal is the host data strobe (HDS) Schmitt-trigger input. The polarity of the data strobe is programmable, but is configured as active-low ($\overline{\text{HDS}}$) following reset.
$\overline{\text{HWR}}$/ HWR	Input		**Host Write Data** — When HDI08 is programmed to interface a double-data-strobe host bus and the HI function is selected, this signal is the host write data strobe (HWR) Schmitt-trigger input. The polarity of the data strobe is programmable, but is configured as active-low ($\overline{\text{HWR}}$) following reset.
PB12	Input, output, or disconnected		**Port B 12** — When the HDI08 is configured as GPIO, this signal is individually programmed as input, output, or internally disconnected.
			The default state after reset for this signal is GPIO disconnected. This input is 5 V tolerant.

Table 2-9 Host Interface (continued)

Signal Name	Type	State during Reset	Signal Description
HCS	Input	GPIO disconnected	**Host Chip Select** — When HDI08 is programmed to interface a nonmultiplexed host bus and the HI function is selected, this signal is the host chip select (HCS) input. The polarity of the chip select is programmable, but is configured active-low ($\overline{\text{HCS}}$) after reset.
HA10	Input		**Host Address 10** — When HDI08 is programmed to interface a multiplexed host bus and the HI function is selected, this signal is line 10 of the host address (HA10) input bus.
PB13	Input, output, or disconnected		**Port B 13** — When the HDI08 is configured as GPIO, this signal is individually programmed as input, output, or internally disconnected.
			The default state after reset for this signal is GPIO disconnected. This input is 5 V tolerant.
$\overline{\text{HOREQ}}$/HOREQ	Output	GPIO disconnected	**Host Request** — When HDI08 is programmed to interface a single host request host bus and the HI function is selected, this signal is the host request (HOREQ) output. The polarity of the host request is programmable, but is configured as active-low ($\overline{\text{HOREQ}}$) following reset. The host request may be programmed as a driven or open-drain output.
$\overline{\text{HTRQ}}$/ HTRQ	Output		**Transmit Host Request** — When HDI08 is programmed to interface a double host request host bus and the HI function is selected, this signal is the transmit host request (HTRQ) output. The polarity of the host request is programmable, but is configured as active-low ($\overline{\text{HTRQ}}$) following reset. The host request may be programmed as a driven or open-drain output.
PB14	Input, output, or disconnected		**Port B 14** — When the HDI08 is configured as GPIO, this signal is individually programmed as input, output, or internally disconnected.
			The default state after reset for this signal is GPIO disconnected. This input is 5 V tolerant.

Table 2-9 Host Interface (continued)

Signal Name	Type	State during Reset	Signal Description
$\overline{\text{HACK}}$/ HACK	Input	GPIO disconnected	**Host Acknowledge** — When HDI08 is programmed to interface a single host request host bus and the HI function is selected, this signal is the host acknowledge (HACK) Schmitt-trigger input. The polarity of the host acknowledge is programmable, but is configured as active-low ($\overline{\text{HACK}}$) after reset.
$\overline{\text{HRRQ}}$/ HRRQ	Output		**Receive Host Request** — When HDI08 is programmed to interface a double host request host bus and the HI function is selected, this signal is the receive host request (HRRQ) output. The polarity of the host request is programmable, but is configured as active-low ($\overline{\text{HRRQ}}$) after reset. The host request may be programmed as a driven or open-drain output.
PB15	Input, output, or disconnected		**Port B 15** — When the HDI08 is configured as GPIO, this signal is individually programmed as input, output, or internally disconnected.
			The default state after reset for this signal is GPIO disconnected. *This input is 5 V tolerant.*

2.8 Serial Host Interface

The SHI has five I/O signals that can be configured to allow the SHI to operate in either SPI or I^2C mode.

Table 2-10 Serial Host Interface Signals

Signal Name	Signal Type	State during Reset	Signal Description
SCK	Input or output	Tri-stated	**SPI Serial Clock** — The SCK signal is an output when the SPI is configured as a master and a Schmitt-trigger input when the SPI is configured as a slave. When the SPI is configured as a master, the SCK signal is derived from the internal SHI clock generator. When the SPI is configured as a slave, the SCK signal is an input, and the clock signal from the external master synchronizes the data transfer. The SCK signal is ignored by the SPI if it is defined as a slave and the slave select ($\overline{\text{SS}}$) signal is not asserted. In both the master and slave SPI devices, data is shifted on one edge of the SCK signal and is sampled on the opposite edge where data is stable. Edge polarity is determined by the SPI transfer protocol.
SCL	Input or output		**I^2C Serial Clock** — SCL carries the clock for I^2C bus transactions in the I^2C mode. SCL is a Schmitt-trigger input when configured as a slave and an open-drain output when configured as a master. SCL should be connected to V_{CC} through a pull-up resistor.
			This signal is tri-stated during hardware, software, and individual reset. Thus, there is no need for an external pull-up in this state. This input is 5 V tolerant.

Table 2-10 Serial Host Interface Signals (continued)

Signal Name	Signal Type	State during Reset	Signal Description
MISO	Input or output	Tri-stated	**SPI Master-In-Slave-Out** — When the SPI is configured as a master, MISO is the master data input line. The MISO signal is used in conjunction with the MOSI signal for transmitting and receiving serial data. This signal is a Schmitt-trigger input when configured for the SPI Master mode, an output when configured for the SPI Slave mode, and tri-stated if configured for the SPI Slave mode when $\overline{SS}$ is deasserted. An external pull-up resistor is not required for SPI operation.
SDA	Input or open-drain output		**I^2C Data and Acknowledge** — In I^2C mode, SDA is a Schmitt-trigger input when receiving and an open-drain output when transmitting. SDA should be connected to V_{CC} through a pull-up resistor. SDA carries the data for I^2C transactions. The data in SDA must be stable during the high period of SCL. The data in SDA is only allowed to change when SCL is low. When the bus is free, SDA is high. The SDA line is only allowed to change during the time SCL is high in the case of start and stop events. A high-to-low transition of the SDA line while SCL is high is a unique situation, and is defined as the start event. A low-to-high transition of SDA while SCL is high is a unique situation defined as the stop event.
			This signal is tri-stated during hardware, software, and individual reset. Thus, there is no need for an external pull-up in this state. This input is 5 V tolerant.
MOSI	Input or output	Tri-stated	**SPI Master-Out-Slave-In** — When the SPI is configured as a master, MOSI is the master data output line. The MOSI signal is used in conjunction with the MISO signal for transmitting and receiving serial data. MOSI is the slave data input line when the SPI is configured as a slave. This signal is a Schmitt-trigger input when configured for the SPI Slave mode.
HA0	Input		**I^2C Slave Address 0** — This signal uses a Schmitt-trigger input when configured for the I^2C mode. When configured for I^2C slave mode, the HA0 signal is used to form the slave device address. HA0 is ignored when configured for the I^2C master mode.
			This signal is tri-stated during hardware, software, and individual reset. Thus, there is no need for an external pull-up in this state. This input is 5 V tolerant.

Table 2-10 Serial Host Interface Signals (continued)

Signal Name	Signal Type	State during Reset	Signal Description
$\overline{SS}$	Input	Tri-stated	**SPI Slave Select** — This signal is an active low Schmitt-trigger input when configured for the SPI mode. When configured for the SPI Slave mode, this signal is used to enable the SPI slave for transfer. When configured for the SPI master mode, this signal should be kept deasserted (pulled high). If it is asserted while configured as SPI master, a bus error condition is flagged. If $\overline{SS}$ is deasserted, the SHI ignores SCK clocks and keeps the MISO output signal in the high-impedance state.
HA2	Input		**I^2C Slave Address 2** — This signal uses a Schmitt-trigger input when configured for the I^2C mode. When configured for the I^2C Slave mode, the HA2 signal is used to form the slave device address. HA2 is ignored in the I^2C master mode.
			This signal is tri-stated during hardware, software, and individual reset. Thus, there is no need for an external pull-up in this state. *This input is 5 V tolerant*.
$\overline{HREQ}$	Input or Output	Tri-stated	**Host Request** — This signal is an active low Schmitt-trigger input when configured for the master mode but an active low output when configured for the slave mode.
			When configured for the slave mode, $\overline{HREQ}$ is asserted to indicate that the SHI is ready for the next data word transfer and deasserted at the first clock pulse of the new data word transfer. When configured for the master mode, $\overline{HREQ}$ is an input. When asserted by the external slave device, it will trigger the start of the data word transfer by the master. After finishing the data word transfer, the master will await the next assertion of $\overline{HREQ}$ to proceed to the next transfer.
			This signal is tri-stated during hardware, software, personal reset, or when the HREQ1–HREQ0 bits in the HCSR are cleared. There is no need for external pull-up in this state. *This input is 5 V toler*ant.

2.9 Enhanced Serial Audio Interface

Table 2-11 Enhanced Serial Audio Interface Signals

Signal Name	Signal Type	State during Reset	Signal Description
HCKR	Input or output	GPIO disconnected	**High Frequency Clock for Receiver** — When programmed as an input, this signal provides a high frequency clock source for the ESAI receiver as an alternate to the DSP core clock. When programmed as an output, this signal can serve as a high-frequency sample clock (e.g., for external digital to analog converters [DACs]) or as an additional system clock.
PC2	Input, output, or disconnected		**Port C 2** — When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. *This input is 5 V tolerant.*
HCKT	Input or output	GPIO disconnected	**High Frequency Clock for Transmitter** — When programmed as an input, this signal provides a high frequency clock source for the ESAI transmitter as an alternate to the DSP core clock. When programmed as an output, this signal can serve as a high frequency sample clock (e.g., for external DACs) or as an additional system clock.
PC5	Input, output, or disconnected		**Port C 5** — When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.
FSR	Input or output	GPIO disconnected	**Frame Sync for Receiver** — This is the receiver frame sync input/output signal. In the asynchronous mode (SYN=0), the FSR pin operates as the frame sync input or output used by all the enabled receivers. In the synchronous mode (SYN=1), it operates as either the serial flag 1 pin (TEBE=0), or as the transmitter external buffer enable control (TEBE=1, RFSD=1). When this pin is configured as serial flag pin, its direction is determined by the RFSD bit in the RCCR register. When configured as the output flag OF1, this pin will reflect the value of the OF1 bit in the SAICR register, and the data in the OF1 bit will show up at the pin synchronized to the frame sync in normal mode or the slot in network mode. When configured as the input flag IF1, the data value at the pin will be stored in the IF1 bit in the SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.
PC1	Input, output, or disconnected		**Port C 1** — When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.

Table 2-11 Enhanced Serial Audio Interface Signals (continued)

Signal Name	Signal Type	State during Reset	Signal Description
FST	Input or output	GPIO disconnected	**Frame Sync for Transmitter** — This is the transmitter frame sync input/output signal. For synchronous mode, this signal is the frame sync for both transmitters and receivers. For asynchronous mode, FST is the frame sync for the transmitters only. The direction is determined by the transmitter frame sync direction (TFSD) bit in the ESAI transmit clock control register (TCCR).
PC4	Input, output, or disconnected		**Port C 4** — When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.
SCKR	Input or output	GPIO disconnected	**Receiver Serial Clock** — SCKR provides the receiver serial bit clock for the ESAI. The SCKR operates as a clock input or output used by all the enabled receivers in the asynchronous mode (SYN=0), or as serial flag 0 pin in the synchronous mode (SYN=1). When this pin is configured as serial flag pin, its direction is determined by the RCKD bit in the RCCR register. When configured as the output flag OF0, this pin will reflect the value of the OF0 bit in the SAICR register, and the data in the OF0 bit will show up at the pin synchronized to the frame sync in normal mode or the slot in network mode. When configured as the input flag IF0, the data value at the pin will be stored in the IF0 bit in the SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.
PC0	Input, output, or disconnected		**Port C 0** — When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.
SCKT	Input or output	GPIO disconnected	**Transmitter Serial Clock** — This signal provides the serial bit rate clock for the ESAI. SCKT is a clock input or output used by all enabled transmitters and receivers in synchronous mode, or by all enabled transmitters in asynchronous mode.
PC3	Input, output, or disconnected		**Port C 3** — When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.

Table 2-11 Enhanced Serial Audio Interface Signals (continued)

Signal Name	Signal Type	State during Reset	Signal Description
SDO5	Output	GPIO disconnected	**Serial Data Output 5** — When programmed as a transmitter, SDO5 is used to transmit data from the TX5 serial transmit shift register.
SDI0	Input		**Serial Data Input 0** — When programmed as a receiver, SDI0 is used to receive serial data into the RX0 serial receive shift register.
PC6	Input, output, or disconnected		**Port C 6** — When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.
SDO4	Output	GPIO disconnected	**Serial Data Output 4** — When programmed as a transmitter, SDO4 is used to transmit data from the TX4 serial transmit shift register.
SDI1	Input		**Serial Data Input 1** — When programmed as a receiver, SDI1 is used to receive serial data into the RX1 serial receive shift register.
PC7	Input, output, or disconnected		**Port C 7** — When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.
SDO3/ SDO3_1	Output	GPIO disconnected	**Serial Data Output 3** — When programmed as a transmitter, SDO3 is used to transmit data from the TX3 serial transmit shift register. When enabled for ESAI_1 operation, this is the ESAI_1 Serial Data Output 3.
SDI2/ SDI2_1	Input		**Serial Data Input 2** — When programmed as a receiver, SDI2 is used to receive serial data into the RX2 serial receive shift register. When enabled for ESAI_1 operation, this is the ESAI_1 Serial Data Input 2.
PC8/PE8	Input, output, or disconnected		**Port C 8** — When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. When enabled for ESAI_1 GPIO, this is the Port E 8 signal. The default state after reset is GPIO disconnected. This input is 5 V tolerant.

Table 2-11 Enhanced Serial Audio Interface Signals (continued)

Signal Name	Signal Type	State during Reset	Signal Description
SDO2/ SDO2_1	Output	GPIO disconnected	**Serial Data Output 2** — When programmed as a transmitter, SDO2 is used to transmit data from the TX2 serial transmit shift register When enabled for ESAI_1 operation, this is the ESAI_1 Serial Data Output 2.
SDI3/ SDI3_1	Input		**Serial Data Input 3** — When programmed as a receiver, SDI3 is used to receive serial data into the RX3 serial receive shift register. When enabled for ESAI_1 operation, this is the ESAI_1 Serial Data Input 3.
PC9/PE9	Input, output, or disconnected		**Port C 9** — When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. When enabled for ESAI_1 GPIO, this is the Port E 9 signal. The default state after reset is GPIO disconnected. This input is 5 V tolerant.
SDO1/ SDO1_1	Output	GPIO disconnected	**Serial Data Output 1** — SDO1 is used to transmit data from the TX1 serial transmit shift register. When enabled for ESAI_1 operation, this is the ESAI_1 Serial Data Output 1.
PC10/PE10	Input, output, or disconnected		**Port C 10** — When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. When enabled for ESAI_1 GPIO, this is the Port E 10 signal. The default state after reset is GPIO disconnected. This input is 5 V tolerant.
SDO0/ SDO0_1	Output	GPIO disconnected	**Serial Data Output 0** — SDO0 is used to transmit data from the TX0 serial transmit shift register. When enabled for ESAI_1 operation, this is the ESAI_1 Serial Data Output 0.
PC11/ PE11	Input, output, or disconnected		**Port C 11** — When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. When enabled for ESAI_1 GPIO, this is the Port E 11 signal. The default state after reset is GPIO disconnected. This input is 5 V tolerant.

2.10 Enhanced Serial Audio Interface_1

Table 2-12 Enhanced Serial Audio Interface_1 Signals

Signal Name	Signal Type	State during Reset	Signal Description
FSR_1	Input or output	GPIO disconnected	**Frame Sync for Receiver_1** — This is the receiver frame sync input/output signal. In the asynchronous mode (SYN=0), the FSR pin operates as the frame sync input or output used by all the enabled receivers. In the synchronous mode (SYN=1), it operates as either the serial flag 1 pin (TEBE=0), or as the transmitter external buffer enable control (TEBE=1, RFSD=1). When this pin is configured as serial flag pin, its direction is determined by the RFSD bit in the RCCR register. When configured as the output flag OF1, this pin will reflect the value of the OF1 bit in the SAICR register, and the data in the OF1 bit will show up at the pin synchronized to the frame sync in normal mode or the slot in network mode. When configured as the input flag IF1, the data value at the pin will be stored in the IF1 bit in the SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.
PE1	Input, output, or disconnected		**Port E 1**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input cannot tolerate 5 V.
FST_1	Input or output	GPIO disconnected	**Frame Sync for Transmitter_1**—This is the transmitter frame sync input/output signal. For synchronous mode, this signal is the frame sync for both transmitters and receivers. For asynchronous mode, FST is the frame sync for the transmitters only. The direction is determined by the transmitter frame sync direction (TFSD) bit in the ESAI transmit clock control register (TCCR).
PE4	Input, output, or disconnected		**Port E 4**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input cannot tolerate 5 V.

Table 2-12 Enhanced Serial Audio Interface_1 Signals (continued)

Signal Name	Signal Type	State during Reset	Signal Description
SCKR_1	Input or output	GPIO disconnected	**Receiver Serial Clock_1** — SCKR provides the receiver serial bit clock for the ESAI. The SCKR operates as a clock input or output used by all the enabled receivers in the asynchronous mode (SYN=0), or as serial flag 0 pin in the synchronous mode (SYN=1). When this pin is configured as serial flag pin, its direction is determined by the RCKD bit in the RCCR register. When configured as the output flag OF0, this pin will reflect the value of the OF0 bit in the SAICR register, and the data in the OF0 bit will show up at the pin synchronized to the frame sync in normal mode or the slot in network mode. When configured as the input flag IF0, the data value at the pin will be stored in the IF0 bit in the SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.
PE0	Input, output, or disconnected		**Port E 0** — When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input cannot tolerate 5 V.
SCKT_1	Input or output	GPIO disconnected	**Transmitter Serial Clock_1** — This signal provides the serial bit rate clock for the ESAI. SCKT is a clock input or output used by all enabled transmitters and receivers in synchronous mode, or by all enabled transmitters in asynchronous mode.
PE3	Input, output, or disconnected		**Port E 3** — When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input cannot tolerate 5 V.
SDO5_1	Output	GPIO disconnected	**Serial Data Output 5_1** — When programmed as a transmitter, SDO5 is used to transmit data from the TX5 serial transmit shift register.
SDI0_1	Input		**Serial Data Input 0_1** — When programmed as a receiver, SDI0 is used to receive serial data into the RX0 serial receive shift register.
PE6	Input, output, or disconnected		**Port E 6** — When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input cannot tolerate 5 V.

Table 2-12 Enhanced Serial Audio Interface_1 Signals (continued)

Signal Name	Signal Type	State during Reset	Signal Description
SDO4_1	Output	GPIO disconnected	**Serial Data Output 4_1** — When programmed as a transmitter, SDO4 is used to transmit data from the TX4 serial transmit shift register.
SDI1_1	Input		**Serial Data Input 1_1** — When programmed as a receiver, SDI1 is used to receive serial data into the RX1 serial receive shift register.
PE7	Input, output, or disconnected		**Port E 7** — When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.

2.11 SPDIF Transmitter Digital Audio Interface

Table 2-13 Digital Audio Interface (DAX) Signals

Signal Name	Type	State During Reset	Signal Description
ACI	Input	GPIO Disconnected	**Audio Clock Input** — This is the DAX clock input. When programmed to use an external clock, this input supplies the DAX clock. The external clock frequency must be 256, 384, or 512 times the audio sampling frequency (256 × Fs, 384 × Fs or 512 × Fs, respectively).
PD0	Input, output, or disconnected		**Port D 0** — When the DAX is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.
ADO	Output	GPIO Disconnected	**Digital Audio Data Output** — This signal is an audio and non-audio output in the form of AES/EBU, CP340 and IEC958 data in a biphase mark format.
PD1	Input, output, or disconnected		**Port D 1** — When the DAX is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.

2.12 Timer

Table 2-14 Timer Signal

Signal Name	Type	State during Reset	Signal Description
TIO0	Input or Output	Input	**Timer 0 Schmitt-Trigger Input/Output** — When timer 0 functions as an external event counter or in measurement mode, TIO0 is used as input. When timer 0 functions in watchdog, timer, or pulse modulation mode, TIO0 is used as output. The default mode after reset is GPIO input. This can be changed to output or configured as a timer input/output through the timer 0 control/status register (TCSR0). If TIO0 is not being used, it is recommended to either define it as GPIO output immediately at the beginning of operation or leave it defined as GPIO input but connected to Vcc through a pull-up resistor in order to ensure a stable logic level at this input. *This input is 5 V tolerant.*

2.13 JTAG/OnCE Interface

Table 2-15 JTAG/OnCE Interface

Signal Name	Signal Type	State during Reset	Signal Description
TCK	Input	Input	**Test Clock** — TCK is a test clock input signal used to synchronize the JTAG test logic. It has an internal pull-up resistor. *This input is 5 V tolerant.*
TDI	Input	Input	**Test Data Input** — TDI is a test data serial input signal used for test instructions and data. TDI is sampled on the rising edge of TCK and has an internal pull-up resistor. This input is 5 V tolerant.
TDO	Output	Tri-stated	**Test Data Output** — TDO is a test data serial output signal used for test instructions and data. TDO is tri-statable and is actively driven in the shift-IR and shift-DR controller states. TDO changes on the falling edge of TCK.
TMS	Input	Input	**Test Mode Select** — TMS is an input signal used to sequence the test controller's state machine. TMS is sampled on the rising edge of TCK and has an internal pull-up resistor. This input is 5 V tolerant.

3 Memory Configuration

3.1 Data and Program Memory Maps

The on-chip memory configuration of the DSP56366 is affected by the state of the CE (Cache Enable), MSW0, MSW1, and MS (Memory Switch) control bits in the OMR register, and by the SC bit in the Status Register. The internal data and program memory configurations are shown in Table 3-1. The address ranges for the internal memory are shown in Table 3-2 and Table 3-3. The memory maps for each memory configuration are shown in Figure 3-1 to Figure 3-16.

Table 3-1 Internal Memory Configurations

Bit Settings					Memory Sizes (24-bit words)							
MSW1	MSW0	CE	MS	SC	Prog RAM	Prog Cache	Prog ROM	Boot ROM	X Data RAM	Y Data RAM	X Data ROM	Y Data ROM
X	X	0	0	0	3K	n.a.	40K	192	13K	7K	32K	8K
X	X	1	0	0	2K	1K	40K	192	13K	7K	32K	8K
0	0	0	1	0	10K	n.a.	40K	192	8K	5K	32K	8K
0	1	0	1	0	8K	n.a.	40K	192	8K	7K	32K	8K
1	0	0	1	0	5K	n.a.	40K	192	11K	7K	32K	8K
0	0	1	1	0	9K	1K	40K	192	8K	5K	32K	8K
0	1	1	1	0	7K	1K	40K	192	8K	7K	32K	8K
1	0	1	1	0	4K	1K	40K	192	11K	7K	32K	8K
X	X	0	0	1	3K	n.a.	n.a.	n.a.	13K	7K	32K	8K
X	X	1	0	1	2K	1K	n.a.	n.a.	13K	7K	32K	8K
0	0	0	1	1	10K	n.a.	n.a.	n.a.	8K	5K	32K	8K
0	1	0	1	1	8K	n.a.	n.a.	n.a.	8K	7K	32K	8K
1	0	0	1	1	5K	n.a.	n.a.	n.a.	11K	7K	32K	8K
0	0	1	1	1	9K	1K	n.a.	n.a.	8K	5K	32K	8K
0	1	1	1	1	7K	1K	n.a.	n.a.	8K	7K	32K	8K
1	0	1	1	1	4K	1K	n.a.	n.a.	11K	7K	32K	8K

Table 3-2 On-chip RAM Memory Locations

Bit Settings					RAM Memory Locations			
MSW1	MSW0	CE	MS	SC	Prog. RAM	Prog. Cache	X Data RAM	Y Data RAM
X	X	0	0	X	$0000 - $0BFF	n.a.	$0000 - $33FF	$0000-$1BFF
X	X	1	0	X	$0000 - $07FF	enabled	$0000 - $33FF	$0000-$1BFF
0	0	0	1	X	$0000 -$27FF	n.a.	$0000 - $1FFF	$0000 - $13FF
0	1	0	1	X	$0000 - $1BFF and $2400 - $27FF	n.a.	$0000 - $1FFF	$0000-$1BFF
1	0	0	1	X	$0000 - $ 0FFF and $2400 - $27FF	n.a.	$0000 - $2BFF	$0000-$1BFF
0	0	1	1	X	$0000 - $23FF	enabled	$0000 - $1FFF	$0000 - $13FF
0	1	1	1	X	$0000 - $1BFF	enabled	$0000 - $1FFF	$0000 - $1BFF
1	0	1	1	X	$0000 - $0FFF	enabled	$0000 - $2BFF	$0000 - $1BFF

Table 3-3 On-chip ROM Memory Locations

Bit Settings					ROM Memory Locations			
MSW1	MSW0	CE	MS	SC	Prog. ROM	Boot. ROM	X Data ROM	Y Data ROM
X	X	X	X	0	$FF1000 - $FFAFFF	$FF0000 - $FF00BF	$004000- $00BFFF	$004000- $005FFF
X	X	X	X	1	no access	no access	$004000- $00BFFF	$004000- $005FFF

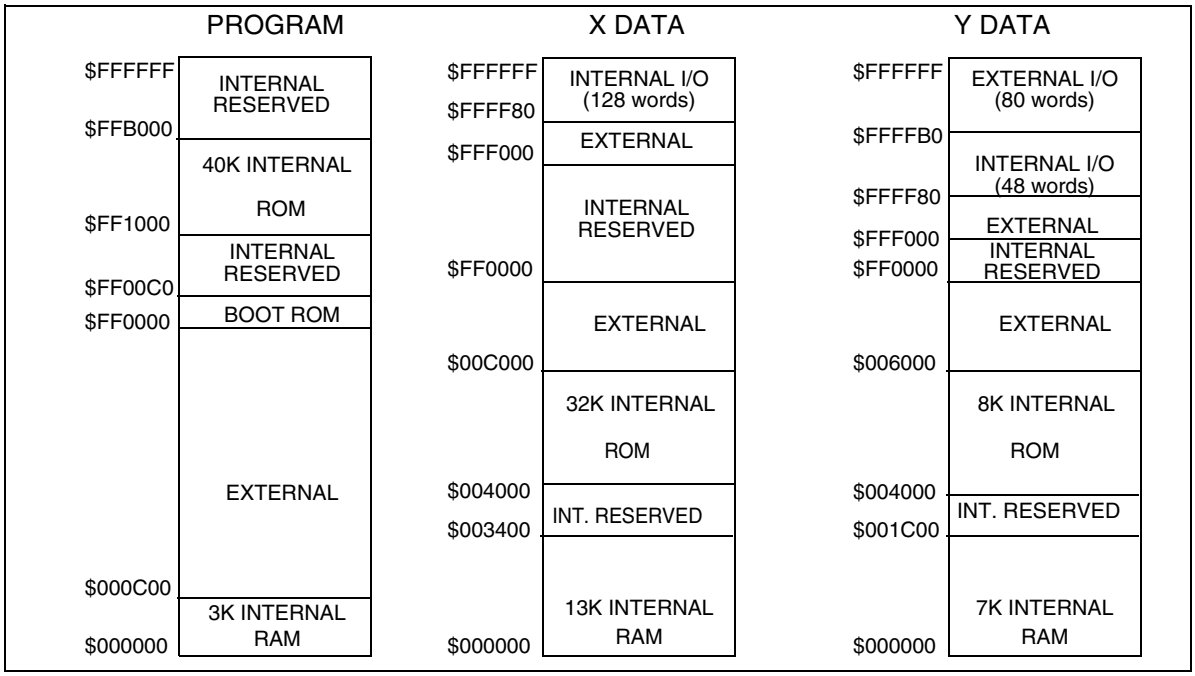


Figure 3-1 Memory Maps for MSW=(X,X), CE=0, MS=0, SC=0

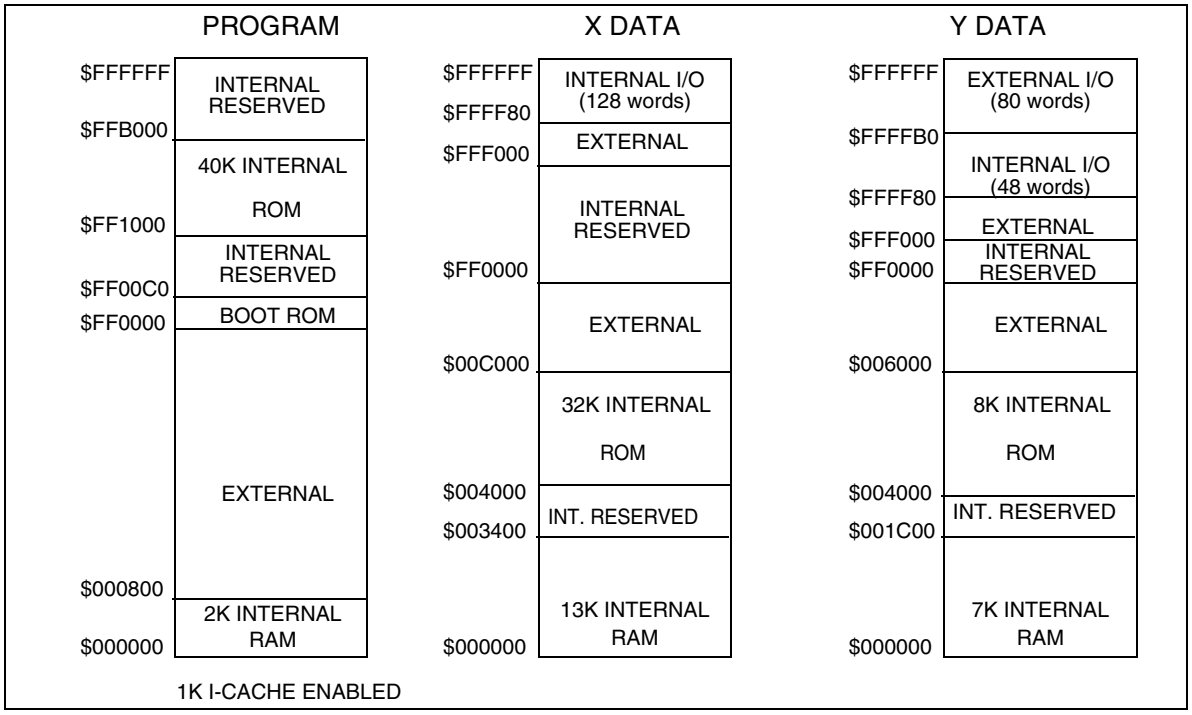


Figure 3-2 Memory Maps for MSW=(X,X), CE=1, MS=0, SC=0

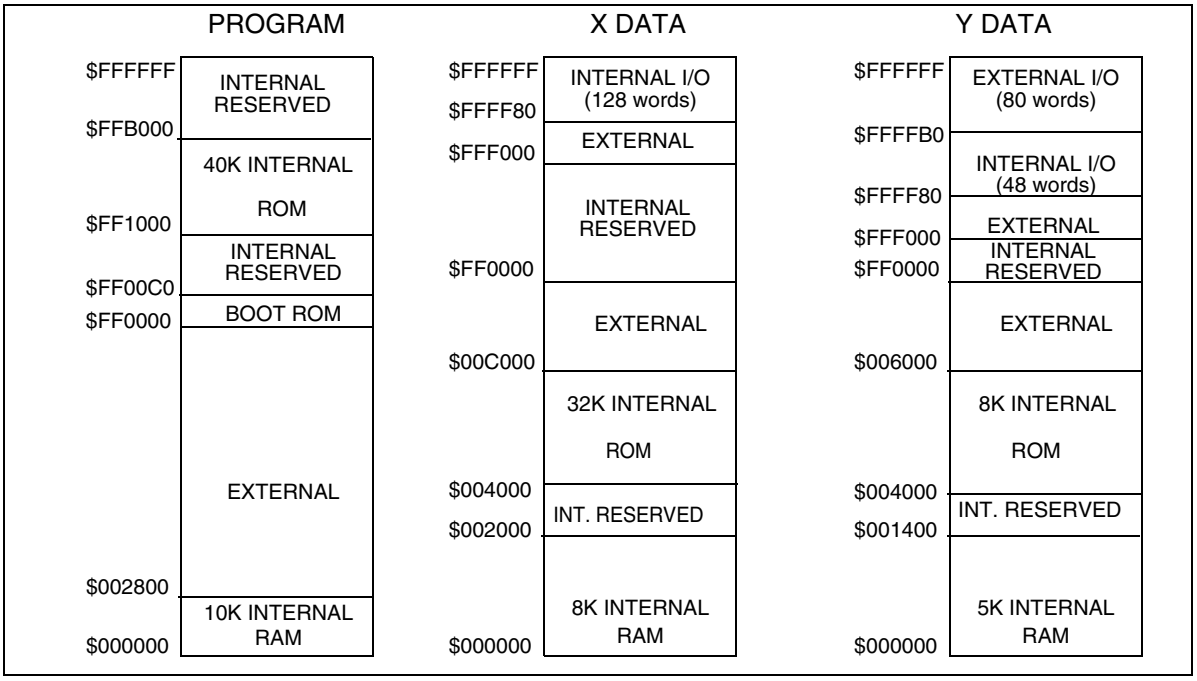


Figure 3-3 Memory Maps for MSW=(0,0), CE=0 MS=1, SC=0

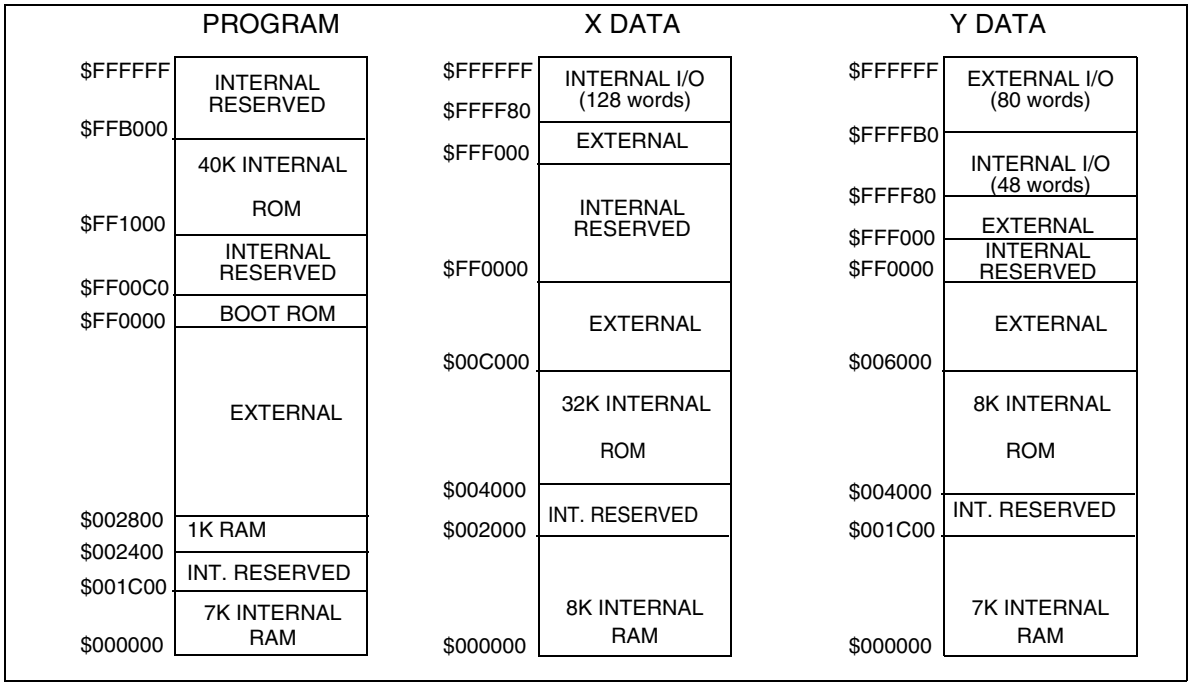


Figure 3-4 Memory Maps for MSW=(0,1), CE=0, MS=1, SC=0

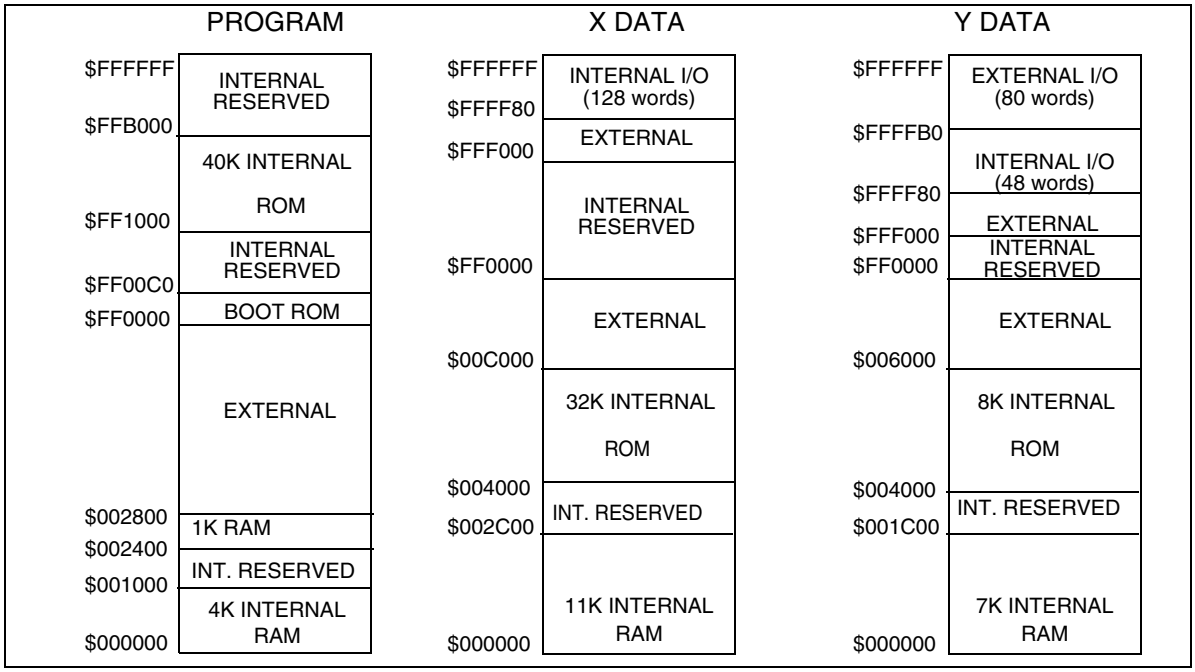


Figure 3-5 Memory Maps for MSW=(1,0), CE=0, MS=1, SC=0

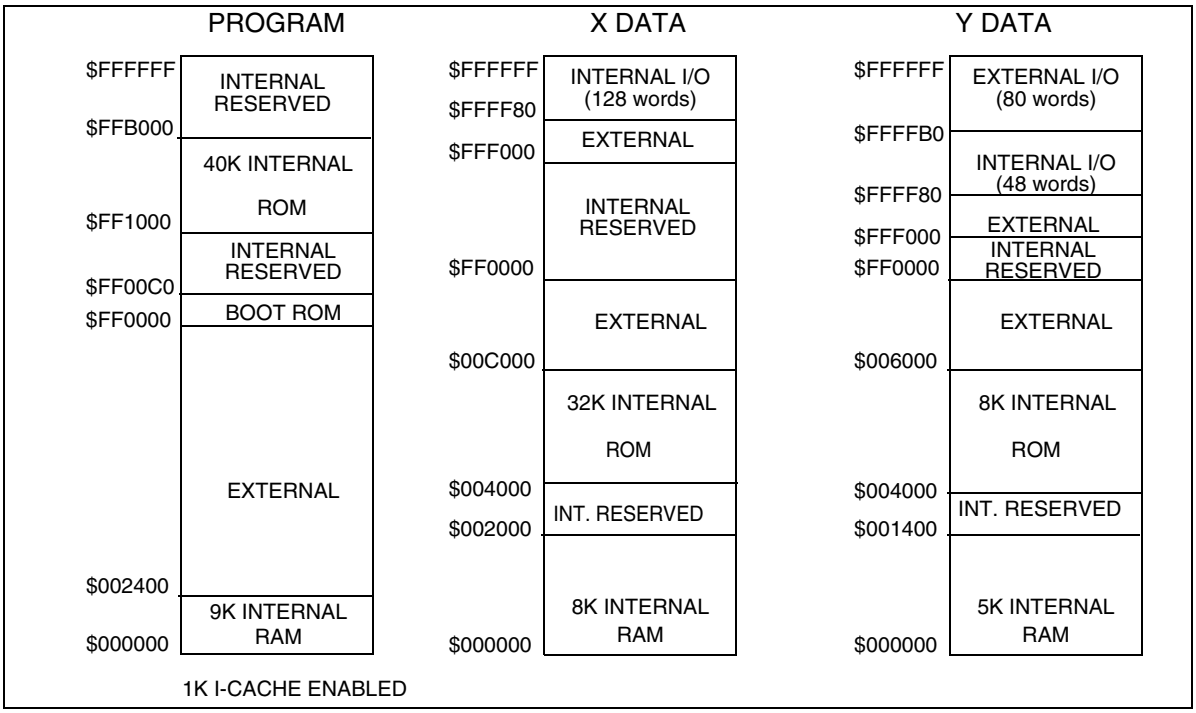


Figure 3-6 Memory Maps for MSW=(0,0), CE=1, MS=1, SC=0

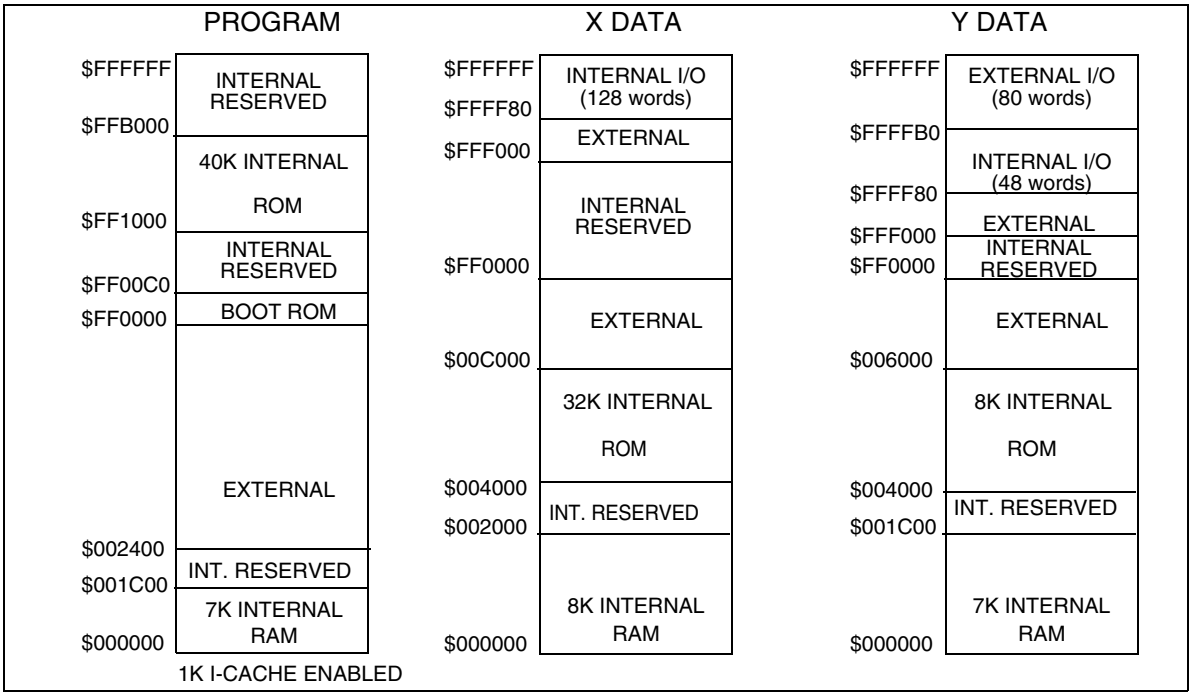


Figure 3-7 Memory Maps for MSW=(0,1), CE=1, MS=1, SC=0

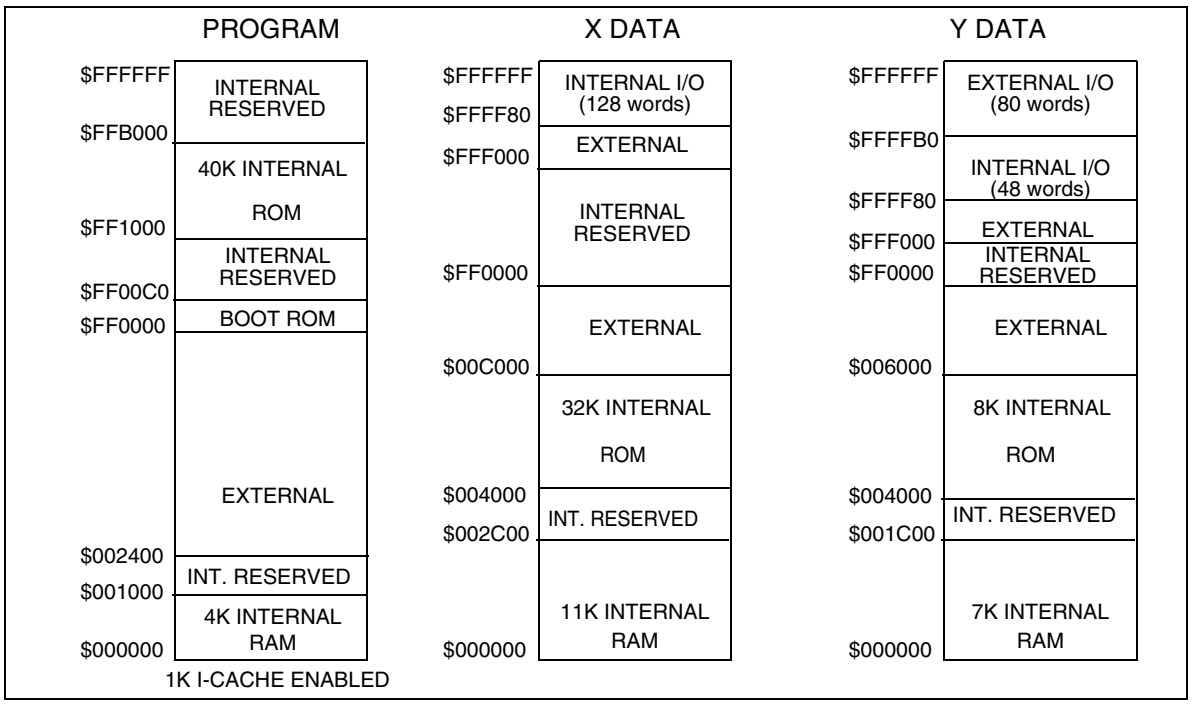


Figure 3-8 Memory Maps for MSW=(1,0), CE=1, MS=1, SC=0

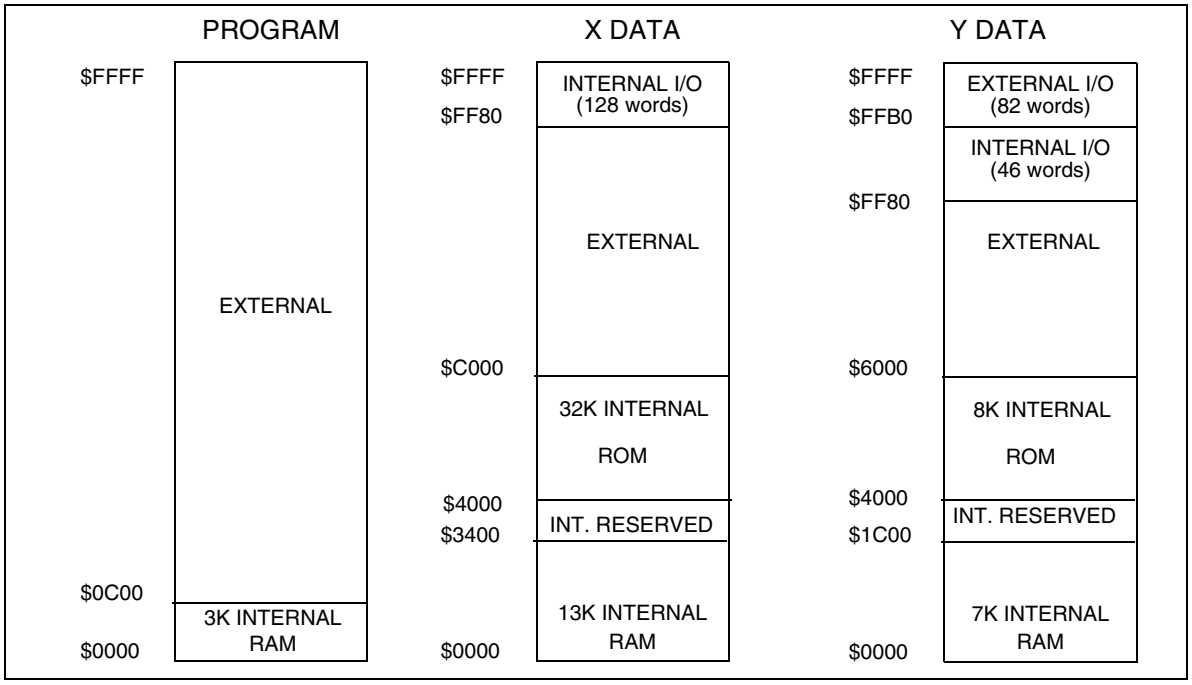


Figure 3-9 Memory Maps for MSW=(X,X), CE=0, MS=0, SC=1

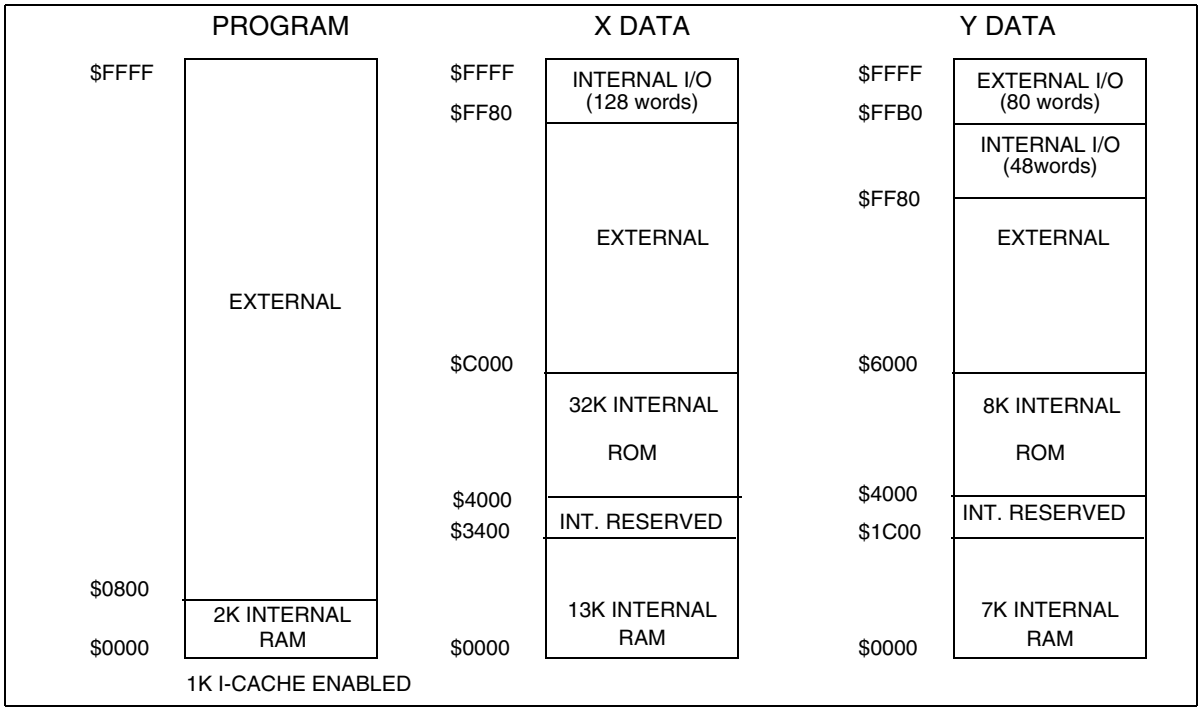


Figure 3-10 Memory Maps for MSW=(X,X), CE=1, MS=0, SC=1

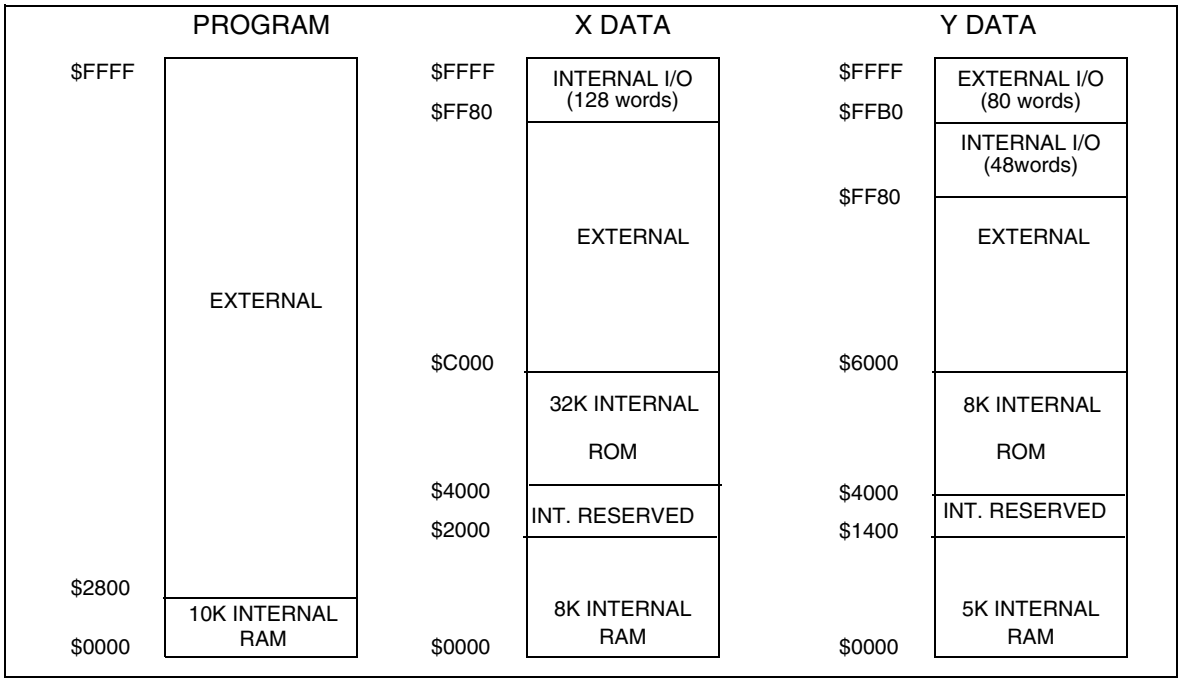


Figure 3-11 Memory Maps for MSW=(0,0), CE=0, MS=1, SC=1

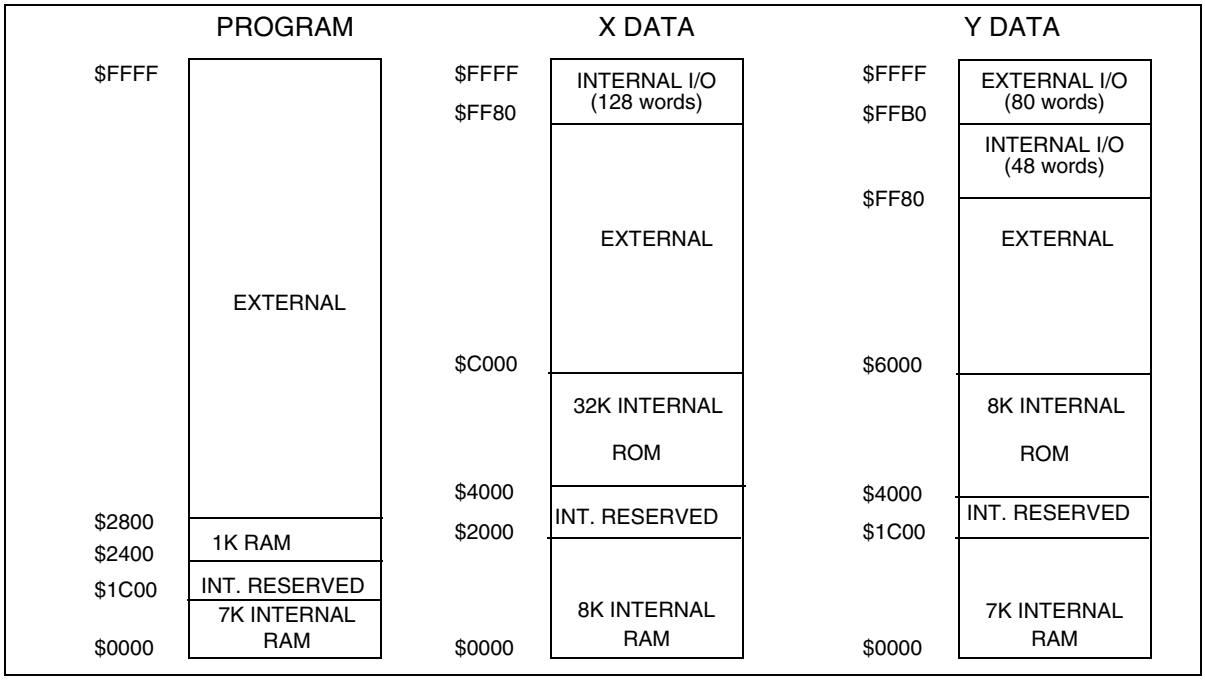


Figure 3-12 Memory Maps for MSW=(0,1), CE=0, MS=1, SC=1

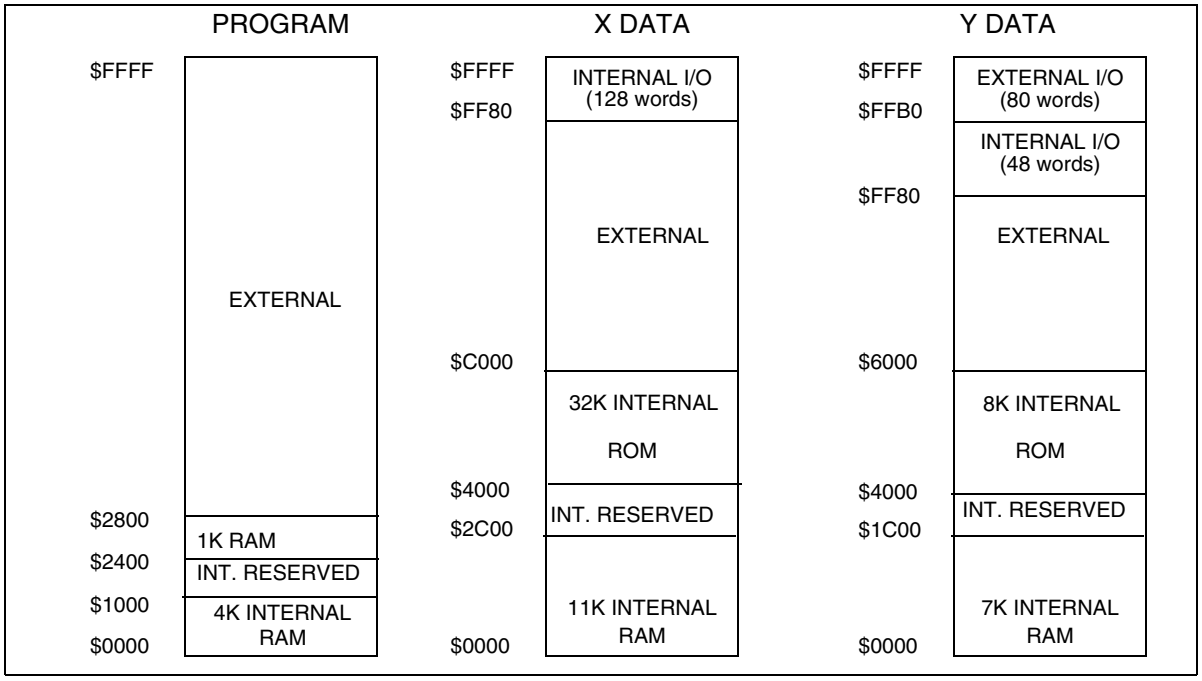


Figure 3-13 Memory Maps for MSW=(1,0), CE=0, MS=1, SC=1

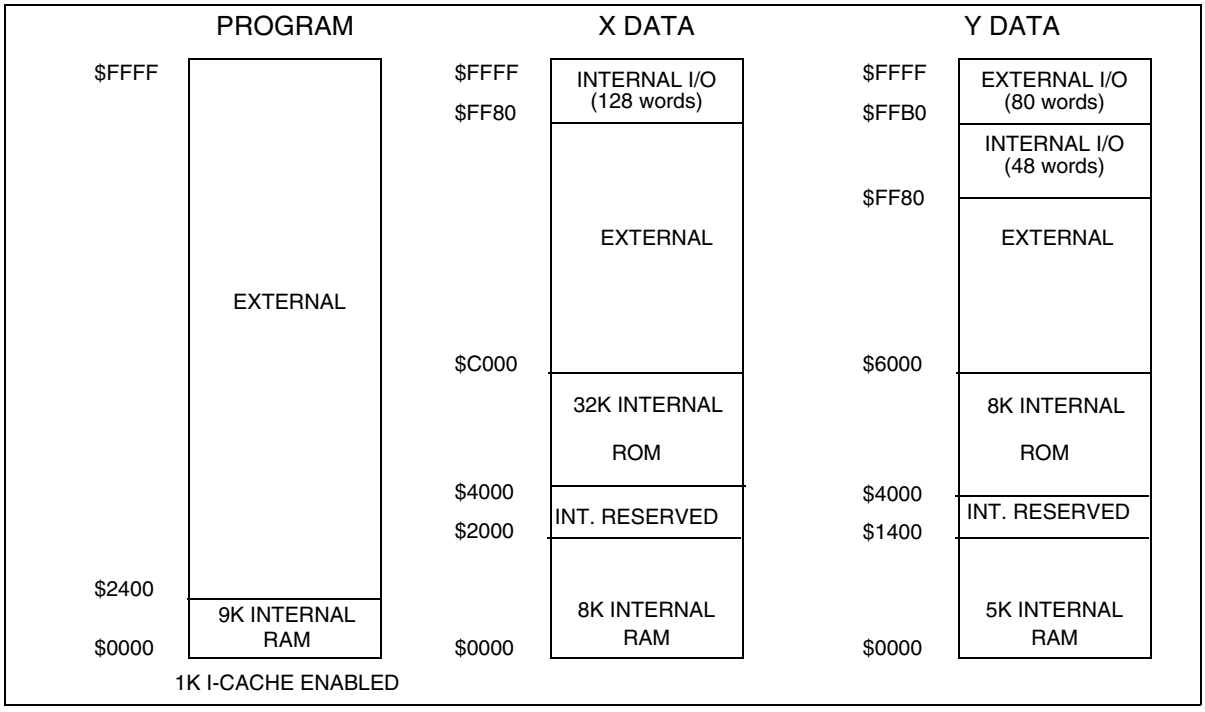


Figure 3-14 Memory Maps for MSW=(0,0), CE=1, MS=1, SC=1

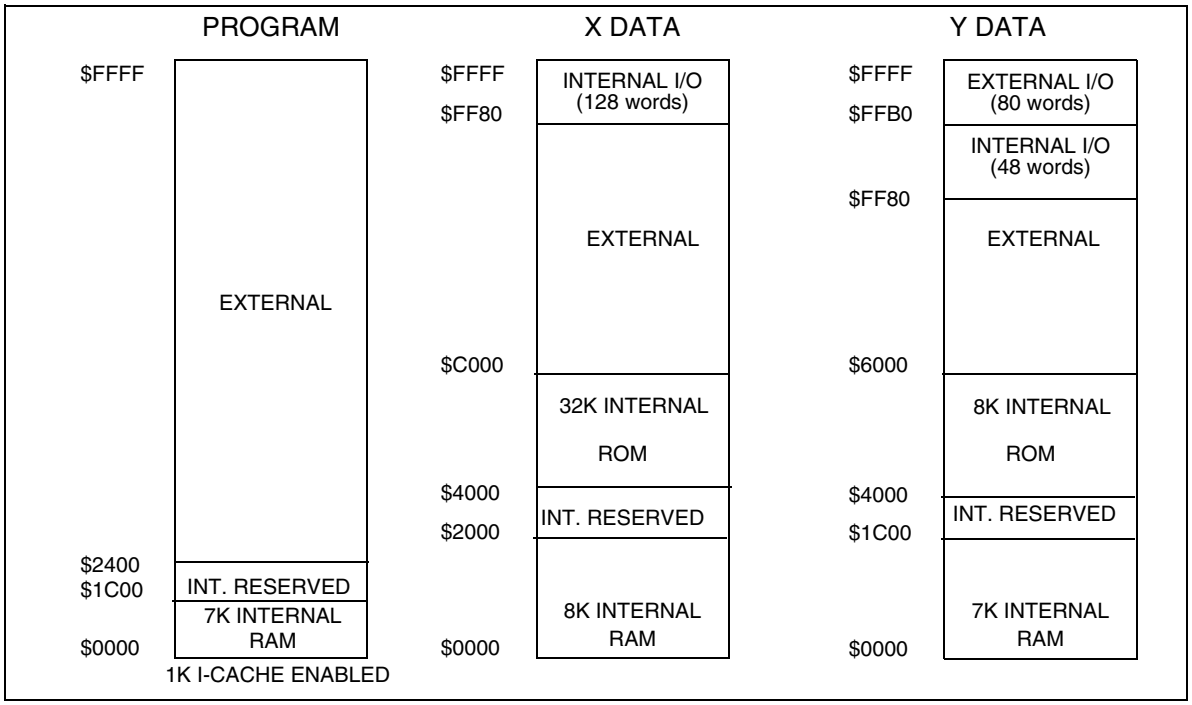


Figure 3-15 Memory Maps for MSW=(0,1), CE=1, MS=1, SC=1

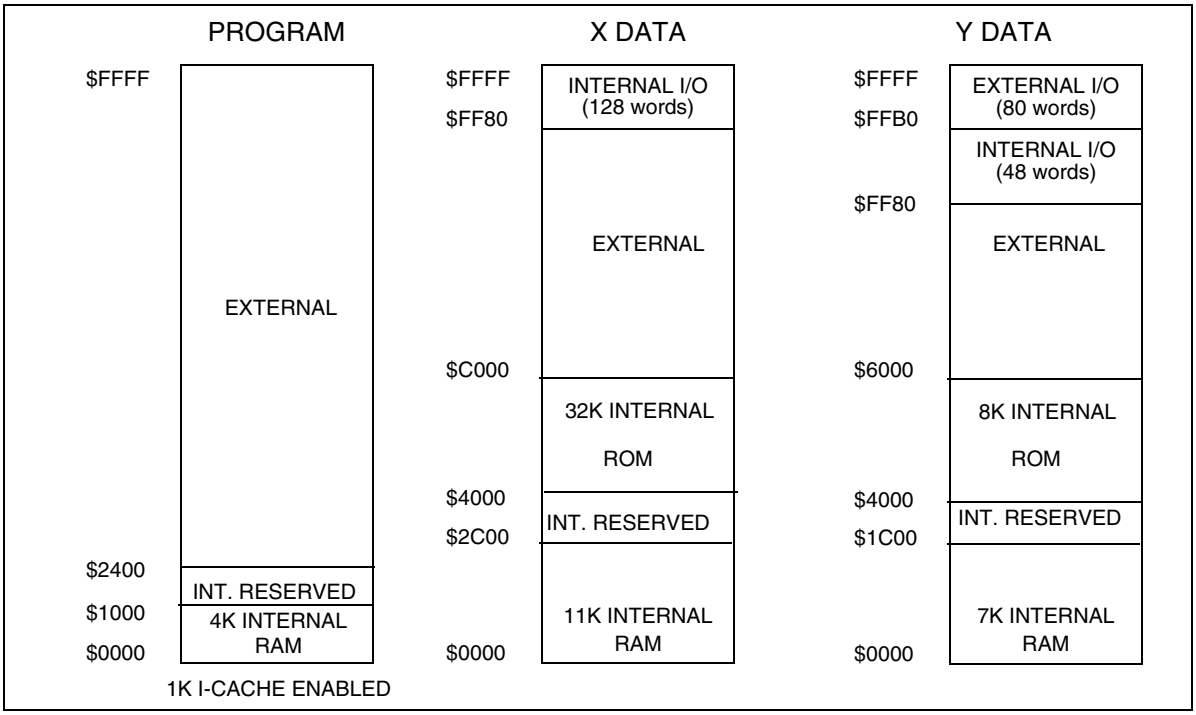


Figure 3-16 Memory Maps for MSW=(1,0), CE=1, MS=1, SC=1

3.1.1 Reserved Memory Spaces

The reserved memory spaces should not be accessed by the user. They are reserved for future expansion.

3.1.2 Program ROM Area Reserved for Freescale Use

The last 128 words ($FFAF80-$FFAFFF) of the Program ROM are reserved for Freescale use. This memory area is reserved for use as expansion area for the bootstrap ROM as well as for testing purposes. Customer code should not use this area. The contents of this Program ROM segment is defined by the Appendix A, "Bootstrap ROM Contents".

3.1.3 Bootstrap ROM

The 192-word Bootstrap ROM occupies locations $FF0000-$FF00BF. The bootstrap ROM is factory-programmed to perform the bootstrap operation following hardware reset. The contents of the Bootstrap ROM are defined by the Bootstrap ROM source code in Appendix A, "Bootstrap ROM Contents".

3.1.4 Dynamic Memory Configuration Switching

The internal memory configuration is altered by re-mapping RAM modules from Y and X data memory into program memory space and vice-versa. The contents of the switched RAM modules are preserved.

The memory can be dynamically switched from one configuration to another by changing the MS, MSW0 or MSW1 bits in OMR. The address ranges that are directly affected by the switch operation are specified in Table 3-2. The memory switch can be accomplished provided that the affected address ranges are not being accessed during the instruction cycle in which the switch operation takes place. Accordingly, the following condition must be observed for trouble-free dynamic switching:

NOTE

No accesses (including instruction fetches) to or from the affected address ranges in program and data memories are allowed during the switch cycle.

NOTE

The switch cycle actually occurs 3 instruction cycles after the instruction that modifies the MS, MSW0 or MSW1 bits.

Any sequence that complies with the switch condition is valid. For example, if the program flow executes in the address range that is not affected by the switch, the switch condition can be met very easily. In this case a switch can be accomplished by just changing the MS, MSW0 or MSW1 bits in OMR in the regular program flow, assuming no accesses to the affected address ranges of the data memory occur up to 3 instructions after the instruction that changes the OMR bit. Special care should be taken in relation to the interrupt vector routines since an interrupt could cause the DSP to fetch instructions out of sequence and might violate the switch condition.

Special attention should be given when running a memory switch routine using the OnCE™ port. Running the switch routine in Trace mode, for example, can cause the switch to complete after the MS bit change

while the DSP is in Debug mode. As a result, subsequent instructions might be fetched according to the new memory configuration (after the switch), and thus might execute improperly.

3.1.5 External Memory Support

The DSP56366 does not support the SSRAM memory type. It does support SRAM and DRAM as indicated in the *DSP56300 24-Bit Digital Signal Processor Family Manual*, Freescale publication DSP56300FM. Also, care should be taken when accessing external memory to ensure that the necessary address lines are available. For example, when using glueless SRAM interfacing, it is possible to directly address 3×2^{18} memory locations (768k) when using the 18 address lines and the three programmable address attribute lines.

3.2 Internal I/O Memory Map

The DSP56366 on-chip peripheral modules have their register files programmed to the addresses in the internal X-I/O memory range (the top 128 locations of the X data memory space) and internal Y-I/O memory range (48 locations of the Y-data memory space) as shown in Table 3-4.

Table 3-4 Internal I/O Memory Map

Peripheral	Address	Register Name
IPR	X:$FFFFFF	INTERRUPT PRIORITY REGISTER CORE (IPR-C)
	X:$FFFFFE	INTERRUPT PRIORITY REGISTER PERIPHERAL (IPR-P)
PLL	X:$FFFFFD	PLL CONTROL REGISTER (PCTL)
ONCE	X:$FFFFFC	ONCE GDB REGISTER (OGDB)
BIU	X:$FFFFFB	BUS CONTROL REGISTER (BCR)
	X:$FFFFFA	DRAM CONTROL REGISTER (DCR)
	X:$FFFFF9	ADDRESS ATTRIBUTE REGISTER 0 (AAR0)
	X:$FFFFF8	ADDRESS ATTRIBUTE REGISTER 1 (AAR1)
	X:$FFFFF7	ADDRESS ATTRIBUTE REGISTER 2 (AAR2)
	X:$FFFFF6	ADDRESS ATTRIBUTE REGISTER 3 (AAR3) [pin not available]
	X:$FFFFF5	ID REGISTER (IDR)
DMA	X:$FFFFF4	DMA STATUS REGISTER (DSTR)
	X:$FFFFF3	DMA OFFSET REGISTER 0 (DOR0)
	X:$FFFFF2	DMA OFFSET REGISTER 1 (DOR1)
	X:$FFFFF1	DMA OFFSET REGISTER 2 (DOR2)
	X:$FFFFF0	DMA OFFSET REGISTER 3 (DOR3)
DMA0	X:$FFFFEF	DMA SOURCE ADDRESS REGISTER (DSR0)
	X:$FFFFEE	DMA DESTINATION ADDRESS REGISTER (DDR0)
	X:$FFFFED	DMA COUNTER (DCO0)
	X:$FFFFEC	DMA CONTROL REGISTER (DCR0)

Table 3-4 Internal I/O Memory Map (continued)

Peripheral	Address	Register Name
DMA1	X:$FFFFEB	DMA SOURCE ADDRESS REGISTER (DSR1)
	X:$FFFFEA	DMA DESTINATION ADDRESS REGISTER (DDR1)
	X:$FFFFE9	DMA COUNTER (DCO1)
	X:$FFFFE8	DMA CONTROL REGISTER (DCR1)
DMA2	X:$FFFFE7	DMA SOURCE ADDRESS REGISTER (DSR2)
	X:$FFFFE6	DMA DESTINATION ADDRESS REGISTER (DDR2)
	X:$FFFFE5	DMA COUNTER (DCO2)
	X:$FFFFE4	DMA CONTROL REGISTER (DCR2)
DMA3	X:$FFFFE3	DMA SOURCE ADDRESS REGISTER (DSR3)
	X:$FFFFE2	DMA DESTINATION ADDRESS REGISTER (DDR3)
	X:$FFFFE1	DMA COUNTER (DCO3)
	X:$FFFFE0	DMA CONTROL REGISTER (DCR3)
DMA4	X:$FFFFDF	DMA SOURCE ADDRESS REGISTER (DSR4)
	X:$FFFFDE	DMA DESTINATION ADDRESS REGISTER (DDR4)
	X:$FFFFDD	DMA COUNTER (DCO4)
	X:$FFFFDC	DMA CONTROL REGISTER (DCR4)
DMA5	X:$FFFFDB	DMA SOURCE ADDRESS REGISTER (DSR5)
	X:$FFFFDA	DMA DESTINATION ADDRESS REGISTER (DDR5)
	X:$FFFFD9	DMA COUNTER (DCO5)
	X:$FFFFD8	DMA CONTROL REGISTER (DCR5)
PORT D	X:$FFFFD7	PORT D CONTROL REGISTER (PCRD)
	X:$FFFFD6	PORT D DIRECTION REGISTER (PRRD)
	X:$FFFFD5	PORT D DATA REGISTER (PDRD)
DAX	X:$FFFFD4	DAX STATUS REGISTER (XSTR)
	X:$FFFFD3	DAX AUDIO DATA REGISTER B (XADRB)
	X:$FFFFD2	DAX AUDIO DATA REGISTER A (XADRA)
	X:$FFFFD1	DAX NON-AUDIO DATA REGISTER (XNADR)
	X:$FFFFD0	DAX CONTROL REGISTER (XCTR)
	X:$FFFFCF	*Reserved*
	X:$FFFFCE	*Reserved*
	X:$FFFFCD	*Reserved*
	X:$FFFFCC	*Reserved*
	X:$FFFFCB	*Reserved*
	X:$FFFFCA	*Reserved*
PORT B	X:$FFFFC9	HOST PORT GPIO DATA REGISTER (HDR)
	X:$FFFFC8	HOST PORT GPIO DIRECTION REGISTER (HDDR)

Table 3-4 Internal I/O Memory Map (continued)

Peripheral	Address	Register Name
HDI08	X:$FFFFC7	HOST TRANSMIT REGISTER (HOTX)
	X:$FFFFC6	HOST RECEIVE REGISTER (HORX)
	X:$FFFFC5	HOST BASE ADDRESS REGISTER (HBAR)
	X:$FFFFC4	HOST PORT CONTROL REGISTER (HPCR)
	X:$FFFFC3	HOST STATUS REGISTER (HSR)
	X:$FFFFC2	HOST CONTROL REGISTER (HCR)
	X:$FFFFC1	*Reserved*
	X:$FFFFC0	*Reserved*
PORT C	X:$FFFFBF	PORT C CONTROL REGISTER (PCRC)
	X:$FFFFBE	PORT C DIRECTION REGISTER (PRRC)
	X:$FFFFBD	PORT C GPIO DATA REGISTER (PDRC)

Table 3-4 Internal I/O Memory Map (continued)

Peripheral	Address	Register Name
ESAI	X:$FFFFBC	ESAI RECEIVE SLOT MASK REGISTER B (RSMB)
	X:$FFFFBB	ESAI RECEIVE SLOT MASK REGISTER A (RSMA)
	X:$FFFFBA	ESAI TRANSMIT SLOT MASK REGISTER B (TSMB)
	X:$FFFFB9	ESAI TRANSMIT SLOT MASK REGISTER A (TSMA)
	X:$FFFFB8	ESAI RECEIVE CLOCK CONTROL REGISTER (RCCR)
	X:$FFFFB7	ESAI RECEIVE CONTROL REGISTER (RCR)
	X:$FFFFB6	ESAI TRANSMIT CLOCK CONTROL REGISTER (TCCR)
	X:$FFFFB5	ESAI TRANSMIT CONTROL REGISTER (TCR)
	X:$FFFFB4	ESAI COMMON CONTROL REGISTER (SAICR)
	X:$FFFFB3	ESAI STATUS REGISTER (SAISR)
	X:$FFFFB2	*Reserved*
	X:$FFFFB1	*Reserved*
	X:$FFFFB0	*Reserved*
	X:$FFFFAF	*Reserved*
	X:$FFFFAE	*Reserved*
	X:$FFFFAD	*Reserved*
	X:$FFFFAC	*Reserved*
	X:$FFFFAB	ESAI RECEIVE DATA REGISTER 3 (RX3)
	X:$FFFFAA	ESAI RECEIVE DATA REGISTER 2 (RX2)
	X:$FFFFA9	ESAI RECEIVE DATA REGISTER 1 (RX1)
	X:$FFFFA8	ESAI RECEIVE DATA REGISTER 0 (RX0)
	X:$FFFFA7	*Reserved*
	X:$FFFFA6	ESAI TIME SLOT REGISTER (TSR)
	X:$FFFFA5	ESAI TRANSMIT DATA REGISTER 5 (TX5)
	X:$FFFFA4	ESAI TRANSMIT DATA REGISTER 4 (TX4)
	X:$FFFFA3	ESAI TRANSMIT DATA REGISTER 3 (TX3)
	X:$FFFFA2	ESAI TRANSMIT DATA REGISTER 2 (TX2)
	X:$FFFFA1	ESAI TRANSMIT DATA REGISTER 1 (TX1)
	X:$FFFFA0	ESAI TRANSMIT DATA REGISTER 0 (TX0)
	X:$FFFF9F	*Reserved*
	X:$FFFF9E	*Reserved*
	X:$FFFF9D	*Reserved*
	X:$FFFF9C	*Reserved*
	X:$FFFF9B	*Reserved*
	X:$FFFF9A	*Reserved*
	X:$FFFF99	*Reserved*
	X:$FFFF98	*Reserved*

Table 3-4 Internal I/O Memory Map (continued)

Peripheral	Address	Register Name
	X:$FFFF97	*Reserved*
	X:$FFFF96	*Reserved*
	X:$FFFF95	*Reserved*
SHI	X:$FFFF94	SHI RECEIVE FIFO (HRX)
	X:$FFFF93	SHI TRANSMIT REGISTER (HTX)
	X:$FFFF92	SHI I^2C SLAVE ADDRESS REGISTER (HSAR)
	X:$FFFF91	SHI CONTROL/STATUS REGISTER (HCSR)
	X:$FFFF90	SHI CLOCK CONTROL REGISTER (HCKR)
TRIPLE TIMER	X:$FFFF8F	TIMER 0 CONTROL/STATUS REGISTER (TCSR0)
	X:$FFFF8E	TIMER 0 LOAD REGISTER (TLR0)
	X:$FFFF8D	TIMER 0 COMPARE REGISTER (TCPR0)
	X:$FFFF8C	TIMER 0 COUNT REGISTER (TCR0)
	X:$FFFF8B	TIMER 1 CONTROL/STATUS REGISTER (TCSR1)
	X:$FFFF8A	TIMER 1 LOAD REGISTER (TLR1)
	X:$FFFF89	TIMER 1 COMPARE REGISTER (TCPR1)
	X:$FFFF88	TIMER 1 COUNT REGISTER (TCR1)
	X:$FFFF87	TIMER 2 CONTROL/STATUS REGISTER (TCSR2)
	X:$FFFF86	TIMER 2 LOAD REGISTER (TLR2)
	X:$FFFF85	TIMER 2 COMPARE REGISTER (TCPR2)
	X:$FFFF84	TIMER 2 COUNT REGISTER (TCR2)
	X:$FFFF83	TIMER PRESCALER LOAD REGISTER (TPLR)
	X:$FFFF82	TIMER PRESCALER COUNT REGISTER (TPCR)
	X:$FFFF81	*Reserved*
	X:$FFFF80	*Reserved*
ESAI MUX PIN CONTROL	Y:$FFFFAF	MUX PIN CONTROL REGISTER (EMUXR)
	Y:$FFFFAE	*Reserved*
	Y:$FFFFAD	*Reserved*

Table 3-4 Internal I/O Memory Map (continued)

Peripheral	Address	Register Name
	Y:$FFFFAC	*Reserved*
	Y:$FFFFAB	*Reserved*
	Y:$FFFFAA	*Reserved*
	Y:$FFFFA9	*Reserved*
	Y:$FFFFA8	*Reserved*
	Y:$FFFFA7	*Reserved*
	Y:$FFFFA6	*Reserved*
	Y:$FFFFA5	*Reserved*
	Y:$FFFFA4	*Reserved*
	Y:$FFFFA3	*Reserved*
	Y:$FFFFA2	*Reserved*
	Y:$FFFFA1	*Reserved*
	Y:$FFFFA0	*Reserved*
PORT E	Y:$FFFF9F	PORT E CONTROL REGISTER (PCRE)
	Y:$FFFF9E	PORT E DIRECTION REGISTER(PPRE)
	Y:$FFFF9D	PORT E GPIO DATA REGISTER(PDRE)

Table 3-4 Internal I/O Memory Map (continued)

Peripheral	Address	Register Name
ESAI_1	Y:$FFFF9C	ESAI_1 RECEIVE SLOT MASK REGISTER B (RSMB_1)
	Y:$FFFF9B	ESAI_1 RECEIVE SLOT MASK REGISTER A (RSMA_1)
	Y:$FFFF9A	ESAI_1 TRANSMIT SLOT MASK REGISTER B (TSMB_1)
	Y:$FFFF99	ESAI_1 TRANSMIT SLOT MASK REGISTER A (TSMA_1)
	Y:$FFFF98	ESAI_1 RECEIVE CLOCK CONTROL REGISTER (RCCR_)
	Y:$FFFF97	ESAI_1 RECEIVE CONTROL REGISTER (RCR_1)
	Y:$FFFF96	ESAI_1 TRANSMIT CLOCK CONTROL REGISTER (TCCR_1)
	Y:$FFFF95	ESAI_1 TRANSMIT CONTROL REGISTER (TCR_1)
	Y:$FFFF94	ESAI_1 COMMON CONTROL REGISTER (SAICR_1)
	Y:$FFFF93	ESAI_1 STATUS REGISTER (SAISR_1)
	Y:$FFFF92	*Reserved*
	Y:$FFFF91	*Reserved*
	Y:$FFFF90	*Reserved*
	Y:$FFFF8F	*Reserved*
	Y:$FFFF8E	*Reserved*
	Y:$FFFF8D	*Reserved*
	Y:$FFFF8C	*Reserved*
	Y:$FFFF8B	ESAI_1 RECEIVE DATA REGISTER 3 (RX3_1)
	Y:$FFFF8A	ESAI_1 RECEIVE DATA REGISTER 2 (RX2_1)
	Y:$FFFF89	ESAI_1 RECEIVE DATA REGISTER 1 (RX1_1)
	Y:$FFFF88	ESAI_1 RECEIVE DATA REGISTER 0 (RX0_1)
	Y:$FFFF87	*Reserved*
	Y:$FFFF86	ESAI_1 TIME SLOT REGISTER (TSR_1)
	Y:$FFFF85	ESAI_1 TRANSMIT DATA REGISTER 5 (TX5_1)
	Y:$FFFF84	ESAI_1 TRANSMIT DATA REGISTER 4 (TX4_1)
	Y:$FFFF83	ESAI_1 TRANSMIT DATA REGISTER 3 (TX3_1)
	Y:$FFFF82	ESAI_1 TRANSMIT DATA REGISTER 2 (TX2_1)
	Y:$FFFF81	ESAI_1 TRANSMIT DATA REGISTER 1 (TX1_1)
	Y:$FFFF80	ESAI_1 TRANSMIT DATA REGISTER 0 (TX0_1)

4 Core Configuration

4.1 Introduction

This chapter contains DSP56300 core configuration information details specific to the DSP56366. These include the following:

- Operating modes
- Bootstrap program
- Interrupt sources and priorities
- DMA request sources
- OMR
- PLL control register
- AA control registers
- JTAG BSR

For more information on specific registers or modules in the DSP56300 core, refer to the *DSP56300 Family Manual* (DSP56300FM).

4.2 Operating Mode Register (OMR)

Refer to the *DSP56300 Family Manual*, Freescale publication DSP56300FM for a description of the OMR bits.

Table 4-1 Operating Mode Register (OMR)

SCS								EOM								COM							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEN	MSW 1 : 0		SEN	WRP	EOV	EUN	XYS	ATE	APD	ABE	BRT	TAS	BE	CDP1:0		MS	SD		EBD	MD	MC	MB	MA

PEN	- Patch Enable	ATE	- Address Tracing Enable	MS	- Master memory Switch Mode
MSW1	- Memory switch mode 1	APD	- Address Priority Disable	SD	- Stop Delay
MSW0	- Memory switch mode 0	ABE	- Asyn. Bus Arbitration Enable		
SEN	- Stack Extension Enable	BRT	- Bus Release Timing	EBD	- External Bus Disable
WRP	- Extended Stack Wrap Flag	TAS	- TA Synchronize Select	MD	- Operating Mode D
EOV	- Extended Stack Overflow Flag	BE	- Burst Mode Enable	MC	- Operating Mode C
EUN	- Extended Stack Underflow Flag	CDP1	- Core-Dma Priority 1	MB	- Operating Mode B
XYS	- Stack Extension Space Select	CDP0	- Core-Dma Priority 0	MA	- Operating Mode A

- Reserved bit. Read as zero, should be written with zero for future compatibility

4.2.1 Asynchronous Bus Arbitration Enable (ABE) - Bit 13

The asynchronous bus arbitration mode is activated by setting the ABE bit in the OMR register. Hardware reset clears the ABE bit.

4.2.2 Address Attribute Priority Disable (APD) - Bit 14

The Address Attribute Priority Disable (APD) bit is used to turn off the address attribute priority mechanism. When this bit is set, more than one address attribute pin AA/$\overline{\text{RAS}}$(2:0) may be simultaneously asserted according to its AAR settings. The APD bit is cleared by hardware reset.

4.2.3 Address Tracing Enable (ATE) - Bit 15

The Address Tracing Enable (ATE) bit is used to turn on Address Tracing (AT) Mode. When the AT Mode is enabled, the DSP56300 Core reflects the addresses of internal fetches and program space moves (MOVEM) to the Address Bus (A0-A17), if the Address Bus is not needed by the DSP56300 Core for external accesses. The ATE bit is cleared on hardware reset.

4.2.4 Patch Enable (PEN) - Bit 23

The Patch Enable function is used for patching Program ROM locations. i.e. to replace during program execution, the contents of the Program ROM. This is done by using the Instruction Cache to supply the instruction word instead of the Program ROM.

The Patch Enable function is activated by setting bit 23 (PEN) in the OMR Register. The PEN bit is cleared by hardware reset.

The Instruction Cache should be initialized with the new instructions according to the following procedure:

These steps should be executed from external memory or by download via host interface:

1. Set Cache Enable = 1
2. Set Patch Enable = 1
3. Initialize TAGs to different values by unlock eight different external sectors
4. Lock the PATCH sector(s)
5. Move new code to locked sector(s), to the addresses that should be replaced
6. Start regular PROM program

```
;*************************************************************************
; PATCH initialization example
;*************************************************************************

                    page    132,55,0,0,0
                    nolist

                    INCLUDE "ioequ.asm"
                    INCLUDE "intequ.asm"

                    list

START               equ     $100                        ; main program starting address
PATCH_OFSET         equ     128                         ; patch offset
M_PAE               equ     23                          ; Patch Enable
M_PROMS             equ     $ffafec             ; ROM area Start
M_PROME             equ     $ffafff             ; ROM area End

                    org     P:START

                    move    #M_PROMS,r0

                    bset    #M_CE,sr                    ; CacheEnable = 1
                    bset    #M_PAE,omr                  ; PatchEnable = 1
                    move    #$800000,r1                 ; any external address
                    move    #128,n1                     ; 128 for 1K ICACHE, sector size
                    move    #(M_PROMS+PATCH_OFSET),r2

                    dup     8
                    punlock (r1)+n1                     ; initialize TAGs to different
                                                        ; values
                    endm

                    plock   (r2)                        ; lock patch's sector
                                                        ; (start/mid/end)

                    move    #PATCH_DATA_START,r1

;
; replace ROM code by PATCH
```

```
;
                    do      #(PATCH_DATA_END-PATCH_DATA_START+1),PATCH_LOOP
                    movem   p:(r1)+,x0
                    movem   x0,p:(r2)+
                    nop                                     ; Do-loop restriction
PATCH_LOOP

                    jsr     #M_PROMS                        ; start ROM code execution

ENDTEST             jmp     ENDTEST
                    nop
                    nop
                    nop
                    nop

;
; patch data
;

PATCH_DATA_START

                    move    #5,m0
                    move    #6,m1
                    move    #7,m2

PATCH_DATA_END

;*************************************************************************
```

4.3 Operating Modes

The operating modes are defined as shown in Table 4-2. The operating modes are latched from MODA, MODB, MODC and MODD pins during reset. Each operating mode is briefly described below. Except for modes 0 and 8, the operation of all other modes is defined by the Bootstrap ROM source code in Appendix A , "Bootstrap ROM Contents".

Table 4-2 DSP56366 Operating Modes

Mode	MOD D	MOD C	MOD B	MOD A	Reset Vector	Description
0	0	0	0	0	$C00000	Expanded mode
1	0	0	0	1	$FF0000	Bootstrap from byte-wide memory
2	0	0	1	0	$FF0000	Jump to PROM starting address
3	0	0	1	1	$FF0000	*Reserved*
4	0	1	0	0	$FF0000	*Reserved*
5	0	1	0	1	$FF0000	Bootstrap from SHI (slave SPI mode)

Table 4-2 DSP56366 Operating Modes (continued)

Mode	MOD D	MOD C	MOD B	MOD A	Reset Vector	Description
6	0	1	1	0	$FF0000	Bootstrap from SHI (slave I^2C mode) (HCKFR=1, 100ns filter enabled)
7	0	1	1	1	$FF0000	Bootstrap from SHI (slave I^2C mode)(HCKR=0)
8	1	0	0	0	$008000	Expanded mode
9	1	0	0	1	$FF0000	*Reserved* for Burn-in testing
A	1	0	1	0	$FF0000	*Reserved*
B	1	0	1	1	$FF0000	*Reserved*
C	1	1	0	0	$FF0000	HDI08 Bootstrap in ISA Mode
D	1	1	0	1	$FF0000	HDI08 Bootstrap in HC11 non-multiplexed mode
E	1	1	1	0	$FF0000	HDI08 Bootstrap in 8051 multiplexed bus mode
F	1	1	1	1	$FF0000	HDI08 Bootstrap in 68302 bus mode

Table 4-3 DSP56366 Mode Descriptions

Mode 0 The DSP starts fetching instructions beginning at address $C00000. Memory accesses are performed using SRAM memory access type with 31 wait states and no address attributes selected. Address $C00000 is reflected as address $00000 on Port A pins A0-A17.

Mode 1 The bootstrap program loads instructions through Port A from external byte-wide memory, connected to the least significant byte of the data bus (bits 7-0), and starting at address P:$D00000. The bootstrap code expects to read 3 bytes specifying the number of program words, 3 bytes specifying the address to start loading the program words and then 3 bytes for each program word to be loaded. The number of words, the starting address and the program words are read least significant byte first followed by the mid and then by the most significant byte. The program words will be stored in contiguous PRAM memory locations starting at the specified starting address. After reading the program words, program execution starts from the same address where loading started.The SRAM memory access type is selected by the values in Address Attribute Register 1 (AAR1), with 31 wait states for each memory access. Address $D00000 is reflected as address $00000 on Port A pins A0-A17.

Mode 2 The DSP starts fetching instructions from the starting address of the on-chip Program ROM.

Mode 3 Reserved.

Mode 4 Reserved.

Mode 5 In this mode, the internal PRAM is loaded from the Serial Host Interface (SHI). The SHI operates in the SPI slave mode, with 24-bit word width.The bootstrap code expects to read a 24-bit word specifying the number of program words, a 24-bit word specifying the address to start loading the program words and then a 24-bit word for each program word to be loaded. The program words will be stored in contiguous PRAM memory locations starting at the specified starting address. After reading the program words, program execution starts from the same address where loading started.

Table 4-3 DSP56366 Mode Descriptions

Mode 6 Same as Mode 5 except SHI interface operates in the I^2C slave mode with HCKFR set to 1 and the 100ns filter enabled.

Mode 7 Same as Mode 5 except SHI interface operates in the I^2C slave mode with HCKFR set to 0.

Mode 8 The DSP starts fetching instructions beginning at address $008000. Memory accesses are performed using SRAM memory access type with 31 wait states and no address attributes selected.

Mode 9 Reserved. Used for Burn-In testing.

Mode A Reserved.

Mode B Reserved.

Mode C Instructions are loaded through the HDI08, which is configured to interface with an ISA bus. The HOST ISA bootstrap code expects to read a 24-bit word specifying the number of program words, a 24-bit word specifying the address to start loading the program words and then a 24-bit word for each program word to be loaded. The program words will be stored in contiguous PRAM memory locations starting at the specified starting address. After reading the program words, program execution starts from the same address where loading started. The Host Interface bootstrap load program may be stopped by setting the Host Flag 0 (HF0). This will start execution of the loaded program from the specified starting address.

Mode D As in Mode C, but HDI08 is set for interfacing to Freescale HC11 microcontroller in non-multiplexed mode

Mode E As in Mode C, but HDI08 is set for interfacing to Intel 8051 multiplexed bus

Mode F As in Mode C, but HDI08 is set for interfacing to Freescale 68302 bus.

4.4 Interrupt Priority Registers

There are two interrupt priority registers in the DSP56366:

1. IPR-C is dedicated for DSP56300 Core interrupt sources.
2. IPR-P is dedicated for DSP56366 peripheral interrupt sources.

The interrupt priority registers are shown in Figure 4-1 and Figure 4-2. The Interrupt Priority Level bits are defined in Table 4-4. The interrupt vectors are shown in Table 4-6 and the interrupt priorities are shown in Table 4-5.

Table 4-4 Interrupt Priority Level Bits

IPL bits		Interrupts Enabled	Interrupt Priority Level
xxL1	xxL0		
0	0	No	—
0	1	Yes	0
1	0	Yes	1
1	1	Yes	2

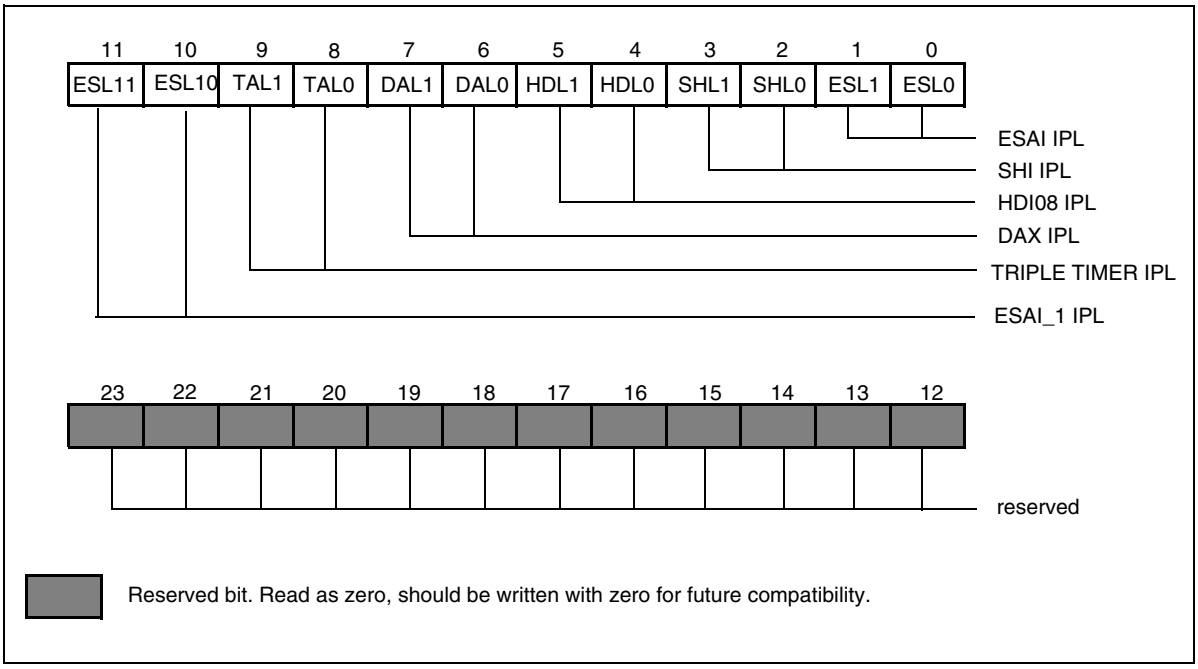

11 10 9 8 7 6 5 4 3 2 1 0
ESL11 ESL10 TAL1 TAL0 DAL1 DAL0 HDL1 HDL0 SHL1 SHL0 ESL1 ESL0
ESAI IPL
SHI IPL
HDI08 IPL
DAX IPL
TRIPLE TIMER IPL
ESAI_1 IPL
23 22 21 20 19 18 17 16 15 14 13 12
reserved
Reserved bit. Read as zero, should be written with zero for future compatibility.

Figure 4-1 Interrupt Priority Register P

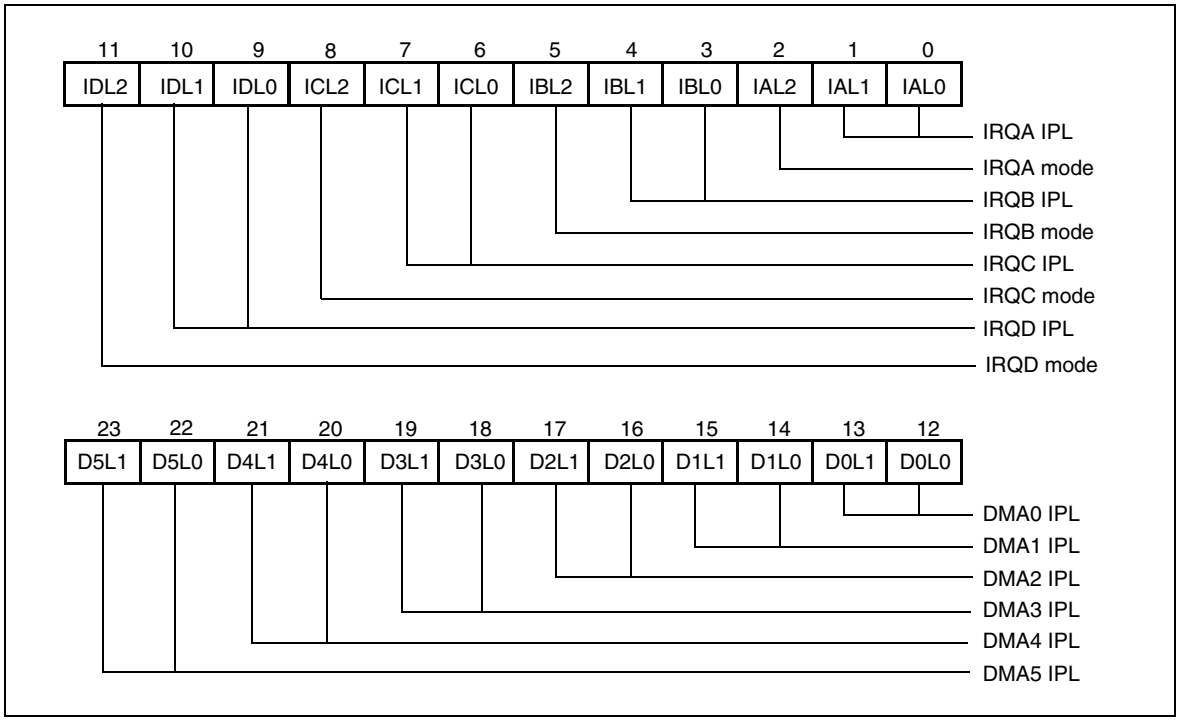

11 10 9 8 7 6 5 4 3 2 1 0
IDL2 IDL1 IDL0 ICL2 ICL1 ICL0 IBL2 IBL1 IBL0 IAL2 IAL1 IAL0
IRQA IPL
IRQA mode
IRQB IPL
IRQB mode
IRQC IPL
IRQC mode
IRQD IPL
IRQD mode
23 22 21 20 19 18 17 16 15 14 13 12
D5L1 D5L0 D4L1 D4L0 D3L1 D3L0 D2L1 D2L0 D1L1 D1L0 D0L1 D0L0
DMA0 IPL
DMA1 IPL
DMA2 IPL
DMA3 IPL
DMA4 IPL
DMA5 IPL

Figure 4-2 Interrupt Priority Register C

Table 4-5 Interrupt Sources Priorities Within an IPL

Priority	Interrupt Source
Level 3 (Nonmaskable)	
Highest	Hardware $\overline{\text{RESET}}$
	Stack Error
	Illegal Instruction
	Debug Request Interrupt
	Trap
Lowest	Non-Maskable Interrupt
Levels 0, 1, 2 (Maskable)	
Highest	$\overline{\text{IRQA}}$ (External Interrupt)
	$\overline{\text{IRQB}}$ (External Interrupt)
	$\overline{\text{IRQC}}$ (External Interrupt)
	$\overline{\text{IRQD}}$ (External Interrupt)
	DMA Channel 0 Interrupt
	DMA Channel 1 Interrupt
	DMA Channel 2 Interrupt
	DMA Channel 3 Interrupt
	DMA Channel 4 Interrupt
	DMA Channel 5 Interrupt
	ESAI Receive Data with Exception Status
	ESAI Receive Even Data
	ESAI Receive Data
	ESAI Receive Last Slot
	ESAI Transmit Data with Exception Status
	ESAI Transmit Last Slot
	ESAI Transmit Even Data
	ESAI Transmit Data
	SHI Bus Error
	SHI Receive Overrun Error
	SHI Transmit Underrun Error

Table 4-5 Interrupt Sources Priorities Within an IPL (continued)

Priority	Interrupt Source
	SHI Receive FIFO Full
	SHI Transmit Data
	SHI Receive FIFO Not Empty
	HOST Command Interrupt
	HOST Receive Data Interrupt
	HOST Transmit Data Interrupt
	DAX Transmit Underrun Error
	DAX Block Transferred
	DAX Transmit Register Empty
	TIMER0 Overflow Interrupt
	TIMER0 Compare Interrupt
	TIMER1 Overflow Interrupt
	TIMER1 Compare Interrupt
	TIMER2 Overflow Interrupt
	TIMER2 Compare Interrupt
	ESAI_1 Receive Data with Exception Status
	ESAI_1 Receive Even Data
	ESAI_1 Receive Data
	ESAI_1 Receive Last Slot
	ESAI_1 Transmit Data with Exception Status
	ESAI_1 Transmit Last Slot
	ESAI_1 Transmit Even Data
Lowest	ESAI_1 Transmit Data

Table 4-6 DSP56366 Interrupt Vectors

Interrupt Starting Address	Interrupt Priority Level Range	Interrupt Source
VBA:$00	3	Hardware $\overline{\text{RESET}}$
VBA:$02	3	Stack Error
VBA:$04	3	Illegal Instruction
VBA:$06	3	Debug Request Interrupt
VBA:$08	3	Trap
VBA:$0A	3	Non-Maskable Interrupt ($\overline{\text{NMI}}$)
VBA:$0C	3	Reserved For Future Level-3 Interrupt Source
VBA:$0E	3	Reserved For Future Level-3 Interrupt Source
VBA:$10	0 - 2	IRQA
VBA:$12	0 - 2	IRQB
VBA:$14	0 - 2	IRQC
VBA:$16	0 - 2	IRQD
VBA:$18	0 - 2	DMA Channel 0
VBA:$1A	0 - 2	DMA Channel 1
VBA:$1C	0 - 2	DMA Channel 2
VBA:$1E	0 - 2	DMA Channel 3
VBA:$20	0 - 2	DMA Channel 4
VBA:$22	0 - 2	DMA Channel 5
VBA:$24	0 - 2	*Reserved*
VBA:$26	0 - 2	*Reserved*
VBA:$28	0 - 2	DAX Underrun Error
VBA:$2A	0 - 2	DAX Block Transferred
VBA:$2C	0 - 2	*Reserved*
VBA:$2E	0 - 2	DAX Audio Data Empty
VBA:$30	0 - 2	ESAI Receive Data
VBA:$32	0 - 2	ESAI Receive Even Data
VBA:$34	0 - 2	ESAI Receive Data With Exception Status
VBA:$36	0 - 2	ESAI Receive Last Slot
VBA:$38	0 - 2	ESAI Transmit Data
VBA:$3A	0 - 2	ESAI Transmit Even Data
VBA:$3C	0 - 2	ESAI Transmit Data with Exception Status
VBA:$3E	0 - 2	ESAI Transmit Last Slot
VBA:$40	0 - 2	SHI Transmit Data
VBA:$42	0 - 2	SHI Transmit Underrun Error

Table 4-6 DSP56366 Interrupt Vectors (continued)

Interrupt Starting Address	Interrupt Priority Level Range	Interrupt Source
VBA:$44	0 - 2	SHI Receive FIFO Not Empty
VBA:$46	0 - 2	*Reserved*
VBA:$48	0 - 2	SHI Receive FIFO Full
VBA:$4A	0 - 2	SHI Receive Overrun Error
VBA:$4C	0 - 2	SHI Bus Error
VBA:$4E	0 - 2	*Reserved*
VBA:$50	0 - 2	*Reserved*
VBA:$52	0 - 2	*Reserved*
VBA:$54	0 - 2	TIMER0 Compare
VBA:$56	0 - 2	TIMER0 Overflow
VBA:$58	0 - 2	TIMER1 Compare
VBA:$5A	0 - 2	TIMER1 Overflow
VBA:$5C	0 - 2	TIMER2 Compare
VBA:$5E	0 - 2	TIMER2 Overflow
VBA:$60	0 - 2	Host Receive Data Full
VBA:$62	0 - 2	Host Transmit Data Empty
VBA:$64	0 - 2	Host Command (Default)
VBA:$66	0 - 2	*Reserved*
VBA:$68	0 - 2	*Reserved*
VBA:$6A	0 - 2	*Reserved*
VBA:$6C	0 - 2	*Reserved*
VBA:$6E	0 - 2	*Reserved*
VBA:$70	0 - 2	ESAI_1 Receive Data
VBA:$72	0 - 2	ESAI_1 Receive Even Data
VBA:$74	0 - 2	ESAI_1 Receive Data With Exception Status
VBA:$76	0 - 2	ESAI_1 Receive Last Slot
VBA:$78	0 - 2	ESAI_1 Transmit Data
VBA:$7A	0 - 2	ESAI_1 Transmit Even Data
VBA:$7C	0 - 2	ESAI_1 Transmit Data with Exception Status
VBA:$7E	0 - 2	ESAI_1 Transmit Last Slot
VBA:$80	0 - 2	*Reserved*
:	:	:
VBA:$FE	0 - 2	*Reserved*

4.5 DMA Request Sources

The DMA Request Source bits (DRS0-DRS4 bits in the DMA Control/Status registers) encode the source of DMA requests used to trigger the DMA transfers. The DMA request sources may be the internal peripherals or external devices requesting service through the $\overline{\text{IRQA}}$, $\overline{\text{IRQB}}$, $\overline{\text{IRQC}}$ and $\overline{\text{IRQD}}$ pins. The DMA Request Sources are shown in Table 4-7.

Table 4-7 DMA Request Sources

DMA Request Source Bits DRS4...DRS0	Requesting Device
00000	External ($\overline{\text{IRQA}}$ pin)
00001	External ($\overline{\text{IRQB}}$ pin)
00010	External ($\overline{\text{IRQC}}$ pin)
00011	External ($\overline{\text{IRQD}}$ pin)
00100	Transfer Done from DMA channel 0
00101	Transfer Done from DMA channel 1
00110	Transfer Done from DMA channel 2
00111	Transfer Done from DMA channel 3
01000	Transfer Done from DMA channel 4
01001	Transfer Done from DMA channel 5
01010	DAX Transmit Data
01011	ESAI Receive Data (RDF=1)
01100	ESAI Transmit Data (TDE=1)
01101	SHI HTX Empty
01110	SHI FIFO Not Empty
01111	SHI FIFO Full
10000	HDI08 Receive Data
10001	HDI08 Transmit Data
10010	TIMER0 (TCF=1)
10011	TIMER1 (TCF=1)
10100	TIMER2 (TCF=1)
10101	ESAI_1 Receive Data (RDF=1)
10110	ESAI_1 Transmit Data (TDE=1)
10111-11111	*Reserved*

4.6 PLL Initialization

4.6.1 PLL Multiplication Factor (MF0-MF11)

The DSP56366 PLL multiplication factor is set to 6 during hardware reset, i.e. the Multiplication Factor Bits MF0-MF11 in the PLL Control Register (PCTL) are set to $005.

4.6.2 PLL Pre-Divider Factor (PD0-PD3)

The DSP56366 PLL Pre-Divider factor is set to 1 during hardware reset, i.e. the Pre-Divider Factor Bits PD0-PD3 in the PLL Control Register (PCTL) are set to $0.

4.6.3 Crystal Range Bit (XTLR)

The Crystal Range (XTLR) bit controls the on-chip crystal oscillator transconductance. The on-chip crystal oscillator is not used on the DSP56366 since no XTAL pin is available. The XTLR bit is set to zero during hardware reset in the DSP56366.

4.6.4 XTAL Disable Bit (XTLD)

The XTAL Disable Bit (XTLD) is set to 1 (XTAL disabled) during hardware reset in the DSP56366.

4.7 Device Identification (ID) Register

The Device Identification Register (IDR) is a 24 bit read only factory programmed register used to identify the different DSP56300 core-based family members. This register specifies the derivative number and revision number. This information may be used in testing or by software. Table 4-8 shows the ID register configuration.

Table 4-8 Identification Register Configuration

23 16	15 12	11 0
Reserved	Revision Number	Derivative Number
$00	$0	$366

4.8 JTAG Identification (ID) Register

The JTAG Identification (ID) Register is a 32 bit, read only thought JTAG, factory programmed register used to distinguish the component on a board according to the IEEE 1149.1 standard. Table 4-9 shows the JTAG ID register configuration.

Table 4-9 JTAG Identification Register Configuration

31 28	27 22	21 12	11 1	0
Version Information	Customer Part Number	Sequence Number	Manufacturer Identity	1
0000	000111	0001001111	00000001110	1

4.9 JTAG Boundary Scan Register (BSR)

The boundary scan register (BSR) in the DSP56366 JTAG implementation contains bits for all device signal and clock pins and associated control signals. All bidirectional pins have a single register bit in the boundary scan register for pin data, and are controlled by an associated control bit in the boundary scan register. The boundary scan register bit definitions are described in Table 4-10.

Table 4-10 DSP56366 BSR Bit Definition

Bit #	Pin Name	Pin Type	BSR Cell Type
0	SDO4_1/SDI1_1	—	Control
1	SDO4_1/SDI1_1	Input/Output	Data
2	IRQA	Input	Data
3	IRQB	Input	Data
4	IRQC	Input	Data
5	IRQD	Input	Data
6	D23	Input/Output	Data
7	D22	Input/Output	Data
8	D21	Input/Output	Data
9	D20	Input/Output	Data
10	D19	Input/Output	Data
11	D18	Input/Output	Data
12	D17	Input/Output	Data
13	D16	Input/Output	Data
14	D15	Input/Output	Data
15	D[23:13]	—	Control
16	D14	Input/Output	Data

Bit #	Pin Name	Pin Type	BSR Cell Type
76	FST_1	—	Control
77	FST_1	Input/Output	Data
78	SDO5_1/SDI0_1		Control
79	SDO5_1/SDI0_1	Input/Output	Data
80	RES	Input	Data
81	HAD0	—	Control
82	HAD0	Input/Output	Data
83	HAD1	—	Control
84	HAD1	Input/Output	Data
85	HAD2	—	Control
86	HAD2	Input/Output	Data
87	HAD3	—	Control
88	HAD3	Input/Output	Data
89	HAD4	—	Control
90	HAD4	Input/Output	Data
91	HAD5	—	Control
92	HAD5	Input/Output	Data

Table 4-10 DSP56366 BSR Bit Definition (continued)

Bit #	Pin Name	Pin Type	BSR Cell Type
17	D13	Input/Output	Data
18	D12	Input/Output	Data
19	D11	Input/Output	Data
20	D10	Input/Output	Data
21	D9	Input/Output	Data
22	D8	Input/Output	Data
23	D7	Input/Output	Data
24	D6	Input/Output	Data
25	D5	Input/Output	Data
26	D4	Input/Output	Data
27	D3	Input/Output	Data
28	D[12:0]	—	Control
29	D2	Input/Output	Data
30	D1	Input/Output	Data
31	D0	Input/Output	Data
32	A17	Output3	Data
33	A16	Output3	Data
34	A15	Output3	Data
35	A[17:9]	—	Control
36	A14	Output3	Data
37	A13	Output3	Data
38	A12	Output3	Data
39	A11	Output3	Data
40	A10	Output3	Data
41	A9	Output3	Data
42	A8	Output3	Data

Bit #	Pin Name	Pin Type	BSR Cell Type
93	HAD6	—	Control
94	HAD6	Input/Output	Data
95	HAD7	—	Control
96	HAD7	Input/Output	Data
97	HAS/A0	—	Control
98	HAS/A0	Input/Output	Data
99	HA8/A1	—	Control
100	HA8/A1	Input/Output	Data
101	HA9/A2	—	Control
102	HA9/A2	Input/Output	Data
103	HCS/A10	—	Control
104	HCS/A10	Input/Output	Data
105	TIO0	—	Control
106	TIO0	Input/Output	Data
107	ACI	—	Control
108	ACI	Input/Output	Data
109	ADO	—	Control
110	ADO	Input/Output	Data
111	HREQ/HTRQ	—	Control
112	HREQ/HTRQ	Input/Output	Data
113	HACK/RRQ	—	Control
114	HACK/RRQ	Input/Output	Data
115	HRW/RD	—	Control
116	HRW/RD	Input/Output	Data
117	HDS/WR	—	Control
118	HDS/WR	Input/Output	Data

Table 4-10 DSP56366 BSR Bit Definition (continued)

Bit #	Pin Name	Pin Type	BSR Cell Type
43	A7	Output3	Data
44	A6	Output3	Data
45	A[8:0]	—	Control
46	A5	Output3	Data
47	A4	Output3	Data
48	A3	Output3	Data
49	A2	Output3	Data
50	A1	Output3	Data
51	A0	Output3	Data
52	BG	Input	Data
53	AA0	—	Control
54	AA0	Output3	Data
55	AA1	—	Control
56	AA1	Output3	Data
57	RD	Output3	Data
58	WR	Output3	Data
59	BB	—	Control
60	BB	Input/Output	Data
61	BR	Output2	Data
62	TA	Input	Data
63	PINIT	Input	Data
64	SCKR_1		Control
65	SCKR_1	Input/Output	Data
66	FSR_1		Control
67	FSR_1	Input/Output	Data
68	$\overline{RD}$,$\overline{WR}$	—	Control
119	HSCKR	—	Control
120	HSCKR	Input/Output	Data
121	HSCKT	—	Control
122	HSCKT	Input/Output	Data
123	SCKR	—	Control
124	SCKR	Input/Output	Data
125	SCKT	—	Control
126	SCKT	Input/Output	Data
127	FSR	—	Control
128	FSR	Input/Output	Data
129	FST	—	Control
130	FST	Input/Output	Data
131	SDO5/SDI0	—	Control
132	SDO5/SDI0	Input/Output	Data
133	SDO4/SDI1	—	Control
134	SDO4/SDI1	Input/Output	Data
135	SDO3/SDI2	—	Control
136	SDO3/SDI2	Input/Output	Data
137	SDO2/SDI3	—	Control
138	SDO2/SDI3	Input/Output	Data
139	SDO1	—	Control
140	SDO1	Input/Output	Data
141	SDO0	—	Control
142	SDO0	Input/Output	Data
143	HREQ	—	Control
144	HREQ	Input/Output	Data

Table 4-10 DSP56366 BSR Bit Definition (continued)

Bit #	Pin Name	Pin Type	BSR Cell Type
69	EXTAL	Input	Data
70	SCKT_1	—	Control
71	SCKT_1	Input/Output	Data
72	CAS	—	Control
73	CAS	Output3	Data
74	AA2	—	Control
75	AA2	Output3	Data

Bit #	Pin Name	Pin Type	BSR Cell Type
145	SS	Input	Data
146	SCK/SCL	—	Control
147	SCK/SCL	Input/Output	Data
148	MISO/SDA	—	Control
149	MISO/SDA	Input/Output	Data
150	MOSI/HA0	—	Control
151	MOSI/HA0	Input/Output	Data

NOTES

5 General Purpose Input/Output

5.1 Introduction

The DSP56362 provides up to 37 bidirectional signals that can be configured as GPIO signals or as peripheral dedicated signals. No dedicated GPIO signals are provided. All of these signals are GPIO by default after reset. The techniques for register programming for all GPIO functionality is very similar between these interfaces. This section describes how signals may be used as GPIO.

5.2 Programming Model

The signals description section of this manual describes the special uses of these signals in detail. There are five groups of these signals which can be controlled separately or as groups:

- Port B: up to 16 GPIO signals (shared with the HDI08 signals)
- Port C: 12 GPIO signals (shared with the ESAI signals)
- Port D: two GPIO signals (shared with the DAX signals)
- Port E: 10 GPIO signals (shared with the ESAI_1 signals)
- Timer: one GPIO signal (shared with the timer/event counter signal)

5.2.1 Port B Signals and Registers

When HDI08 is disabled, all 16 HDI08 signals can be used as GPIO. When HDI08 is enabled, five (HA8, HA9, HCS, HOREQ, and HACK) of the 16 port B signals, if not used as a HDI08 signal, can be configured as GPIO signals. The GPIO functionality of port B is controlled by three registers: host port control register (HPCR), host port GPIO data register (HDR), and host port GPIO direction register (HDDR). These registers are described in Section 6, "Host Interface (HDI08)" of this document.

5.2.2 Port C Signals and Registers

Each of the 12 port C signals not used as an ESAI signal can be configured individually as a GPIO signal. The GPIO functionality of port C is controlled by three registers: port C control register (PCRC), port C direction register (PRRC), and port C data register (PDRC). These registers are described in Section 8, "Enhanced Serial AUDIO Interface (ESAI)".

5.2.3 Port D Signals and Registers

Each of the two Port D signals not used as a DAX signal can be configured individually as a GPIO signal. The GPIO functionality of Port D is controlled by three registers: Port D control register (PCRD), Port D direction register (PRRD) and Port D data register (PDRD). These registers are described in Section 10, "Digital Audio Transmitter".

5.2.4 Port E Signals and Registers

Port E has 10 signals, shared with the ESAI_1. Six of the ESAI_1 signals have their own pin, so each of the six signals, if not used as an ESAI_1 signal, can be configured individually as a GPIO signal. The other four ESAI_1 signals share pins with the ESAI. For these shared pins, if the pin is not being used by the ESAI, Port C and the ESAI_1, then it may be used as a Port E GPIO signal. The GPIO functionality of port E is controlled by three registers: port E control register (PCRE), port E direction register (PRRE), and port E data register (PDRE). These registers are described in Section 9, "Enhanced Serial Audio Interface 1 (ESAI_1)".

5.2.5 Timer/Event Counter Signals

The timer/event counter signal (TIO), when not used as a timer signal can be configured as a GPIO signal. The signal is controlled by the appropriate timer control status register (TCSR). The register is described in Section 11, "Timer/ Event Counter".

6 Host Interface (HDI08)

6.1 Introduction

The host interface (HDI08) is a byte-wide, full-duplex, double-buffered, parallel port that can be connected directly to the data bus of a host processor. The HDI08 supports a variety of buses and provides glueless connection with a number of industry standard microcomputers, microprocessors, DSPs and DMA hardware.

The host bus can operate asynchronously to the DSP core clock, therefore the HDI08 registers are divided into 2 banks. The host register bank is accessible to the external host and the DSP register bank is accessible to the DSP core.

The HDI08 supports three classes of interfaces:

- Host processor/Microcontroller (MCU) connection interface
- DMA controller interface
- General purpose I/O (GPIO) port

6.2 HDI08 Features

6.2.1 Interface - DSP side

- Mapping:
  - Registers are directly mapped into eight internal X data memory locations
- Data Word:
  - 24-bit (native) data words are supported, as are 8-bit and 16-bit words
- Transfer Modes:
  - DSP to Host
  - Host to DSP
  - Host Command
- Handshaking Protocols:
  - Software polled
  - Interrupt driven
  - Core DMA accesses
- Instructions:
  - Memory-mapped registers allow the standard MOVE instruction to be used to transfer data between the DSP and the external host.
  - Special MOVEP instruction provides for I/O service capability using fast interrupts.

— Bit addressing instructions (e.g. BCHG, BCLR, BSET, BTST, JCLR, JSCLR, JSET, JSSET) simplify I/O service routines.

6.2.2 Interface - Host Side

- Sixteen signals are provided to support non-multiplexed or multiplexed buses:
  — H0-H7/HAD0-HAD7
  Host data bus (H7-H0) **or** host multiplexed address/data bus (HAD0-HAD7)
  — HAS/HA0
  Address strobe (HAS) **or** Host address line HA0
  — HA8/HA1
  Host address line HA8 **or** Host address line HA1
  — HA9/HA2
  Host address line HA9 **or** Host address line HA2
  — HRW/HRD
  Read/write select (HRW) **or** Read Strobe (HRD)
  — HDS/HWR
  Data Strobe (HDS) **or** Write Strobe (HWR)
  — HCS/HA10
  Host chip select (HCS) **or** Host address line HA10
  — HOREQ/HTRQ
  Host request (HOREQ) **or** Host transmit request (HTRQ)
  — HACK/HRRQ
  Host acknowledge (HACK) **or** Host receive request (HRRQ)
- Mapping:
  — HDI08 registers are mapped into eight consecutive byte locations in the external host bus address space.
  — The HDI08 acts as a memory or IO-mapped peripheral for microprocessors, microcontrollers, etc.
- Data Word:
  — 8-bit
- Transfer Modes:
  — Mixed 8-bit, 16-bit and 24-bit data transfers
    – DSP to Host
    – Host to DSP
  — Host Command
- Handshaking Protocols:
  — Software polled
  — Interrupt-driven (Interrupts are compatible with most processors, including the MC68000, 8051, HC11 and Hitachi H8).

  - Cycle-stealing DMA with initialization
- Dedicated Interrupts:
  - Separate interrupt lines for each interrupt source
  - Special host commands force DSP core interrupts under host processor control, which are useful for the following:
    - Real-Time Production Diagnostics
    - Debugging Window for Program Development
    - Host Control Protocols
- Interface Capabilities:
  - Glueless interface (no external logic required) to the following:
    - Freescale HC11
    - Hitachi H8
    - 8051 family
    - Thomson P6 family
    - external DMA controllers
  - Minimal glue-logic (pullups, pulldowns) required to interface to the following:
    - ISA bus
    - Motorola 68K family
    - Intel X86 family.

6.3 HDI08 Host Port Signals

The host port signals are described in Section 2, "Signal/Connection Descriptions". If the Host Interface functionality is not required, the 16 pins may be defined as general purpose I/O pins PB0-PB15. When the HDI08 is in use, only five host port signals (HA8, HA9, HCS, HOREQ and HACK) may be individually programmed as GPIO pins if they are not needed for their HDI08 function. Summary of the HDI08 signals.

Table 6-1 HDI08 Signal Summary

HDI08 Port Pin	Multiplexed address/data bus Mode	Non Multiplexed bus Mode	GPIO Mode
HAD0-HAD7	HAD0-HAD7	H0-H7	PB0-PB7
HAS/HA0	HAS/$\overline{\text{HAS}}$	HA0	PB8
HA8/HA1	HA8	HA1	PB9
HA9/HA2	HA9	HA2	PB10
HCS/HA10	HA10	HCS/$\overline{\text{HCS}}$	PB13

Table 6-2 Strobe Signals Support signals

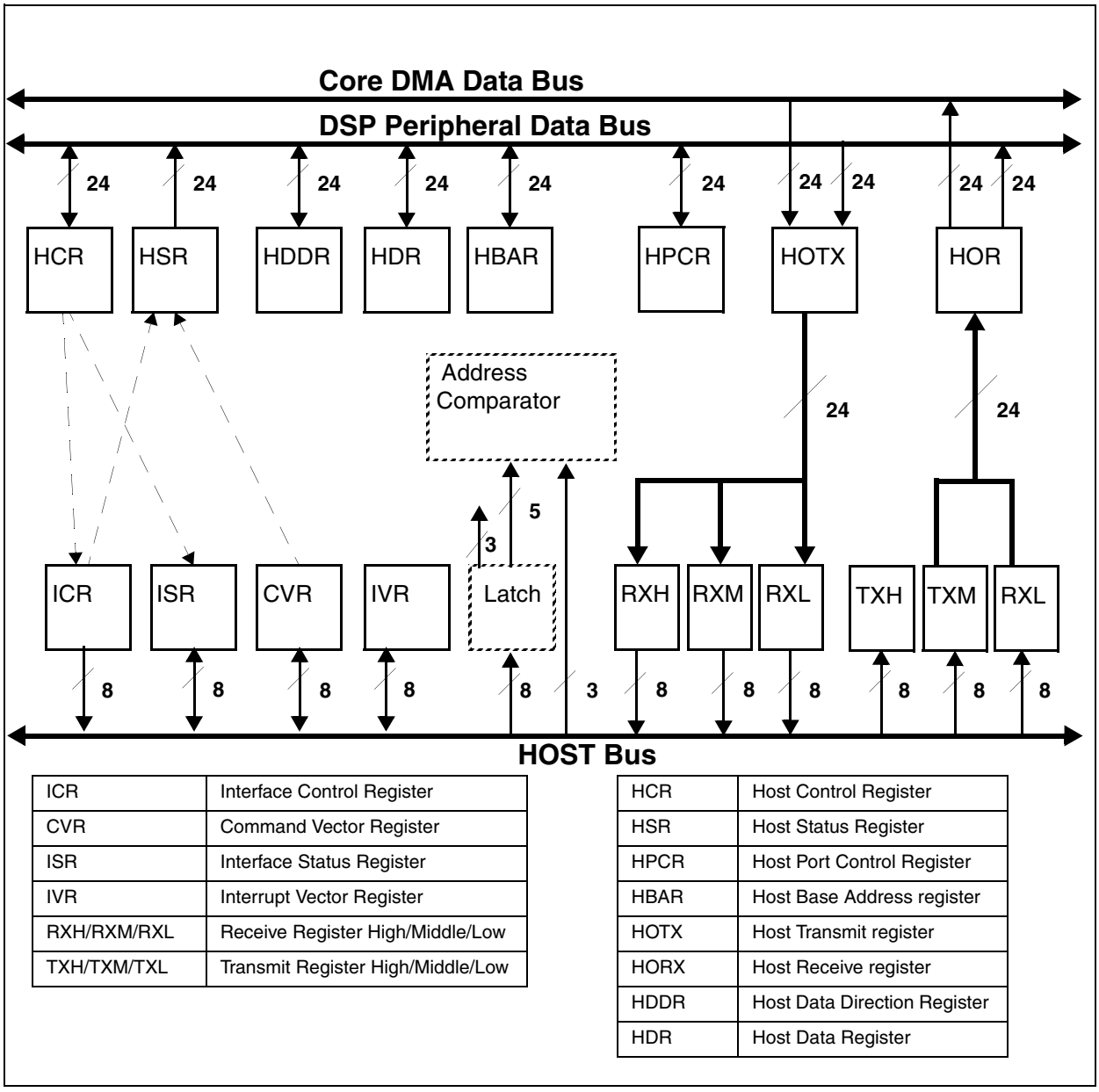
HDI08 Port Pin	Single strobe bus	Dual strobe bus	GPIO Mode
HRW/HRD	HRW	HRD/$\overline{\text{HRD}}$	PB11
HDS/HWR	HDS/$\overline{\text{HDS}}$	HWR/$\overline{\text{HWR}}$	PB12

Table 6-3 Host request support signals

HDI08 Port Pin	Vector required	No vector required	GPIO Mode
HOREQ/HTRQ	HOREQ/$\overline{\text{HOREQ}}$	HTRQ/$\overline{\text{HTRQ}}$	PB14
HACK/HRRQ	HACK/$\overline{\text{HACK}}$	HRRQ/$\overline{\text{HRRQ}}$	PB15

6.4 HDI08 Block Diagram

Figure 6-1 shows the HDI08 registers. The top row of registers (HCR, HSR, HDDR, HDR, HBAR, HPCR, HOTX, HORX) can be accessed the DSP core. The bottom row of registers (ISR, ICR, CVR, IVR, RXH:RXM:RXL and TXH:TXM:TXL) can be accessed by the host processor.



ICR	Interface Control Register
CVR	Command Vector Register
ISR	Interface Status Register
IVR	Interrupt Vector Register
RXH/RXM/RXL	Receive Register High/Middle/Low
TXH/TXM/TXL	Transmit Register High/Middle/Low

HCR	Host Control Register
HSR	Host Status Register
HPCR	Host Port Control Register
HBAR	Host Base Address register
HOTX	Host Transmit register
HORX	Host Receive register
HDDR	Host Data Direction Register
HDR	Host Data Register

Figure 6-1 HDI08 Block Diagram

6.5 HDI08 – DSP-Side Programmer's Model

The DSP core threats the HDI08 as a memory-mapped peripheral occupying eight 24-bit words in X data memory space. The DSP may use the HDI08 as a normal memory-mapped peripheral, employing either standard polled or interrupt-driven programming techniques. Separate transmit and receive data registers are double-buffered to allow the DSP and host processor to transfer data efficiently at high speed. Direct memory mapping allows the DSP core to communicate with the HDI08 registers using standard instructions and addressing modes. In addition, the MOVEP instruction allows direct data transfers between the DSP memory and the HDI08 registers or vice-versa. The HOTX and HORX registers may be serviced by the on-chip DMA controller for data transfers.

The eight host processor registers consists of two data registers and six control registers. All registers can be accessed by the DSP core but not by the external processor.

Data registers are 24-bit registers used for high-speed data transfer to and from the DSP. They are as follows:

- Host Data Receive Register (HORX)
- Host Data Transmit Register (HOTX)

The control registers are 16-bit registers used to control the HDI08 functions. The eight MSBs in the control registers are read by the DSP as zero. The control registers are as follows:

- Host control register (HCR)
- Host status register (HSR)
- Host base address register (HBAR)
- Host port control register (HPCR)
- Host GPIO data direction register (HDDR)
- Host GPIO data register (HDR)

Hardware and software reset disable the HDI08. After reset, the HDI08 signals are configured as GPIO with all pins disconnected.

6.5.1 Host Receive Data Register (HORX)

The 24-bit read-only HORX register is used for host-to-DSP data transfers. The HORX register is loaded with 24-bit data from the transmit data registers (TXH:TXM:TXL) on the host side when both the transmit data register empty TXDE (host side) and host receive data full HRDF (DSP side) bits are cleared. This transfer operation sets both the TXDE and HRDF flags. The HORX register contains valid data when the HRDF bit is set. Reading HORX clears HRDF. The DSP may program the HRIE bit to cause a host receive data interrupt when HRDF is set. Also, a DMA channel may be programmed to read the HORX when HRDF is set.

6.5.2 Host Transmit Data Register (HOTX)

The 24-bit write-only HOTX register is used for DSP- to-host data transfers. Writing to the HOTX register clears the host transfer data empty flag HTDE (DSP side). The contents of the HOTX register are transferred as 24-bit data to the receive byte registers (RXH:RXM:RXL) when both the HTDE flag (DSP side) and receive data full RXDF flag (host side) are cleared. This transfer operation sets the RXDF and HTDE flags. The DSP may set the HTIE bit to cause a host transmit data interrupt when HTDE is set. Also, a DMA Channel may be programmed to write to HOTX when HTDE is set. To prevent the previous data from being overwritten, data should not be written to the HOTX until the HTDE flag is set.

NOTE

When writing data to a peripheral device, there is a two-cycle pipeline delay until any status bits affected by the operation are updated. If the programmer reads any of those status bits within the next two cycles, the bit will not reflect its current status. See the *DSP56300 Family Manual* , Freescale publication DSP56300FM for further details.

6.5.3 Host Control Register (HCR)

The HCR is 16-bit read/write control register used by the DSP core to control the HDI08 operating mode. The initialization values for the HCR bits are described in Section 6.5.9, "DSP-Side Registers After Reset". The HCR bits are described in the following paragraphs.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HDM2	HMD1	HDM0	HF3	HF2	HCIE	HTIE	HRIE

- Reserved bit. Read as 0. Should be written with 0 for future compatibility.

Figure 6-2 Host Control Register (HCR) (X:$FFFFC2)

6.5.3.1 HCR Host Receive Interrupt Enable (HRIE) Bit 0

The HRIE bit is used to enable the host receive data interrupt request. When the host receive data full (HRDF) status bit in the host status register (HSR) is set, a host receive data interrupt request occurs if HRIE is set. If HRIE is cleared, HRDF interrupts are disabled.

6.5.3.2 HCR Host Transmit Interrupt Enable (HTIE) Bit 1

The HTIE bit is used to enable the host transmit data empty interrupt request. When the host transmit data empty (HTDE) status bit in the HSR is set, a host transmit data interrupt request occurs if HTIE is set. If HTIE is cleared, HTDE interrupts are disabled.

6.5.3.3 HCR Host Command Interrupt Enable (HCIE) Bit 2

The HCIE bit is used to enable the host command interrupt request. When the host command pending (HCP) status bit in the HSR is set, a host command interrupt request occurs if HCIE is set. If HCIE is cleared, HCP interrupts are disabled. The interrupt address is determined by the host command vector register (CVR).

NOTE

Host interrupt request priorities: If more than one interrupt request source is asserted and enabled (e.g. HRDF=1, HCP=1, HRIE=1 and HCIE=1), the HDI08 generates interrupt requests according to the following table:

Table 6-4 HDI08 IRQ

Priority	Interrupt Source
Highest	Host Command (HCP=1)
	Transmit Data (HTDE=1)
Lowest	Receive Data (HRDF=1)

6.5.3.4 HCR Host Flags 2,3 (HF2,HF3) Bits 3-4

HF2 and HF3 bits are used as a general-purpose flags for DSP to host communication. HF2 and HF3 may be set or cleared by the DSP core. HF2 and HF3 are reflected in the interface status register (ISR) on the host side such that if they are modified by the DSP software, the host processor can read the modified values by reading the ISR.

These two flags are not designated for any specific purpose but are general-purpose flags. They can be used individually or as encoded pairs in a simple DSP to host communication protocol, implemented in both the DSP and the host processor software.

6.5.3.5 HCR Host DMA Mode Control Bits (HDM0, HDM1, HDM2) Bits 5-7

The HDM[2:0] bits are used to enable the HDI08 DMA mode operation. The HDI08 DMA mode supports external DMA controller devices connected to the HDI08 on the Host side. This mode should not be confused with the operation of the on-chip DMA controller.

With HDM[2:0] cleared, the HDI08 does not support DMA mode operation and the TREQ and RREQ control bits are used for host processor interrupt control via the external HOREQ output signal (or HRREQ and HTREQ output signals if HDREQ in the ICR is set). Also, in the non-DMA mode, the HACK input signal is used for the MC68000 Family vectored interrupt acknowledge input. If HDM[2:0] are not all cleared, the HDI08 operates as described in Table 6-5.

Table 6-5 HDM[2:0] Functionality

HDM			Mode	
2	**1**	**0**	**Description**	**ICR**
0	0	0	DMA operation disabled	INIT | | HLEND | HF1 | HF0 | HDRQ | TREQ | RREQ
1	0	0	DMA Operation Enabled. Host may set HM1 or HM0 in the ICR to enable DMA transfers.	INIT | HM1 | HM0 | HF1 | HF0 | | TREQ | RREQ

Table 6-5 HDM[2:0] Functionality (continued)

HDM			Mode	
2	**1**	**0**	**Description**	**ICR**
0	0	1	DMA Mode Data Output Transfers Enabled. (24-Bit words)	
0	1	0	DMA Mode Data Output Transfers Enabled. (16-Bit words)	
0	1	1	DMA Mode Data Output Transfers Enabled. (8-Bit words)	
1	0	1	DMA Mode Data Input Transfers Enabled. (24-Bit words)	
1	1	0	DMA Mode Data Input Transfers Enabled. (16-Bit words)	
1	1	1	DMA Mode Data Input Transfers Enabled. (8-Bit words)	

INIT	HDM1	HDM0	HF1	HF0		TREQ	RREQ

If HDM1 or HDM0 are set, the DMA mode is enabled, and the HOREQ signal is used to request DMA transfers (the value of the HM1, HM0, HLEND and HDREQ bits in the ICR have no affect). When the DMA mode is enabled, the HDM2 bit selects the direction of DMA transfers:

- setting HDM2 sets the direction of DMA transfer to be DSP to host and enables the HOREQ signal to request data transfer.
- clearing HDM2 sets the direction of DMA transfer to be host to DSP and enables the HOREQ signal to request data transfer.

The HACK input signal is used as a DMA transfer acknowledge input. If the DMA direction is from DSP to host, the contents of the selected register are driven onto the host data bus when HACK is asserted. If the DMA direction is from host to DSP, the selected register is written from the host data bus when HACK is asserted.

The size of the DMA word to be transferred is determined by the DMA control bits, HDM[1:0]. Only the data registers TXH, TXM, TXL and RXH, RXM, RXL can be accessed in DMA mode.The HDI08 data register selected during a DMA transfer is determined by a 2-bit address counter, which is preloaded with the value in HDM[1:0]. The address counter substitutes for the address bits of the HDI08 during a DMA transfer. The address counter can be initialized with the INIT bit feature. After each DMA transfer on the host data bus, the address counter is incremented to the next register. When the address counter reaches the highest register (RXL or TXL), the address counter is not incremented but is loaded with the value in HDM[1:0]. This allows 8-, 16- or 24-bit data to be transferred in a circular fashion and eliminates the need

for the DMA controller to supply the HA2, HA1, and HA0 signals. For 16- or 24-bit data transfers, the DSP CPU interrupt rate is reduced by a factor of 2 or 3, respectively, from the host request rate – i.e., for every two or three host processor data transfers of one byte each, there is only one 24-bit DSP CPU interrupt.

If HDM1 or HDM0 are set, the HM[1:0] bits in the ICR register reflect the value of HDM[1:0].

The HDM[2:0] bits should be changed only while HEN is cleared in the HPCR.

6.5.3.6 HCR Reserved Bits 8-15

These bits are reserved. They read as zero and should be written with zero for future compatibility.

6.5.4 Host Status Register (HSR)

The HSR is a 16-bit read-only status register used by the DSP to read the status and flags of the HDI08. It cannot be directly accessed by the host processor. The initialization values for the HSR bits are described in Section 6.5.9, "DSP-Side Registers After Reset". The HSR bits are described in the following paragraphs.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								DMA			HF1	HF0	HCP	HTDE	HRDF

- Reserved bit. Read as 0. Should be written with 0 for future compatibility.

Figure 6-3 Host Status Register (HSR) (X:FFFFC3)

6.5.4.1 HSR Host Receive Data Full (HRDF) Bit 0

The HRDF bit indicates that the host receive data register (HORX) contains data from the host processor. HRDF is set when data is transferred from the TXH:TXM:TXL registers to the HORX register. HRDF is cleared when HORX is read by the DSP core. If HRDF is set the HDI08 generates a receive data full DMA request, if enabled by a DSP core DMA Channel. If HRDF is set when HRIE is set, a host receive data interrupt request is generated. HRDF can also be cleared by the host processor using the initialize function.

6.5.4.2 HSR Host Transmit Data Empty (HTDE) Bit 1

The HTDE bit indicates that the host transmit data register (HOTX) is empty and can be written by the DSP core. HTDE is set when the HOTX register is transferred to the RXH:RXM:RXL registers. HTDE is cleared when HOTX is written by the DSP core. If HTDE is set the HDI08 generates a transmit data empty DMA request, if enabled by a DSP core DMA Channel. If HTDE is set when HTIE is set, a host transmit data interrupt request is generated. HTDE can also be set by the host processor using the initialize function.

6.5.4.3 HSR Host Command Pending (HCP) Bit 2

The HCP bit indicates that the host has set the HC bit and that a host command interrupt is pending. The HCP bit reflects the status of the HC bit in the command vector register (CVR). HC and HCP are cleared

by the HDI08 hardware when the interrupt request is serviced by the DSP core. The host can clear HC, which also clears HCP.

6.5.4.4 HSR Host Flags 0,1 (HF0,HF1) Bits 3-4

HF0 and HF1 bits are used as a general-purpose flags for host to DSP communication. HF0 and HF1 may be set or cleared by the host. HF0 and HF1 reflect the status of host flags HF0 and HF1 in the ICR register on the host side.

These two flags are not designated for any specific purpose but are general-purpose flags. They can be used individually or as encoded pairs in a simple host to DSP communication protocol, implemented in both the DSP and the host Processor software.

6.5.4.5 HSR Reserved Bits 5-6, 8-15

These bits are reserved. They read as zero and should be written with zero for future compatibility.

6.5.4.6 HSR DMA Status (DMA) Bit 7

The DMA status bit is set when the DMA mode of operation is enabled, and is cleared when the DMA mode is disabled. The DMA mode is enabled under the following conditions:

- HCR bits HDM[2:0] = 100 and the host processor has enabled the DMA mode by setting either or both the ICR bits HM1 and HM0
- Either or both of the HCR bits HDM1 and HDM0 have been set

When the DMA bit is zero, the channel not in use can be used for polled or interrupt operation by the DSP.

6.5.5 Host Base Address Register (HBAR)

The HBAR is used in multiplexed bus modes. This register selects the base address where the host side registers are mapped into the bus address space. The address from the host bus is compared with the base address as programmed in the base address register. If the addresses match, an internal chip select is generated. The use of this register by the chip selectlogic is shown in Figure 6-5.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								BA10	BA9	BA8	BA7	BA6	BA5	BA4	BA3

- Reserved bit. Read as 0. Should be written with 0, for future compatibility.

Figure 6-4 Host Base Address Register (HBAR) (X:$FFFFC5)

6.5.5.1 HBAR Base Address (BA[10:3]) Bits 0-7

These bits define the base address where the host side registers are mapped into the bus address space.

6.5.5.2 HBAR Reserved Bits 8-15

These bits are reserved. They read as zero and should be written with zero for future compatibility.

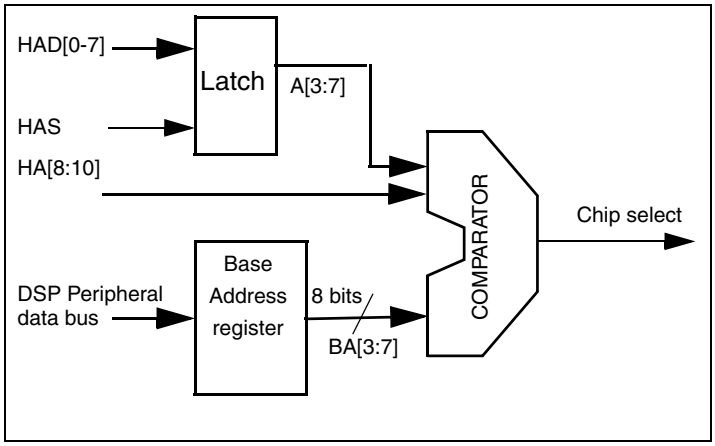


Figure 6-5 Self Chip Select logic

6.5.6 Host Port Control Register (HPCR)

The HPCR is a 16-bit read/write control register used by the DSP to control the HDI08 operating mode. The initialization values for the HPCR bits are described in Section 6.5.9, "DSP-Side Registers After Reset". The HPCR bits are described in the following paragraphs.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HAP	HRP	HCSP	HDDS	HMUX	HASP	HDSP	HROD		HEN	HAEN	HREN	HCSEN	HA9EN	HA8EN	HGEN

- Reserved bit. Read as 0. Should be written with 0, for future compatibility.

Figure 6-6 Host Port Control Register (HPCR) (X:$FFFFC4)

NOTE

To assure proper operation of the HDI08, the HPCR bits HAP, HRP, HCSP, HDDS, HMUX, HASP, HDSP, HROD, HAEN and HREN should be changed only if HEN is cleared. Also, the HPCR bits HAP, HRP, HCSP, HDDS, HMUX, HASP, HDSP, HROD, HAEN, HREN, HCSEN, HA9EN and HA8EN should not be set when HEN is set or simultaneously with setting HEN.

6.5.6.1 HPCR Host GPIO Port Enable (HGEN) Bit 0

If the HGEN bit is set, pins configured as GPIO are enabled. If this bit is cleared, pins configured as GPIO are disconnected: outputs are high impedance, inputs are electrically disconnected. Pins configured as HDI08 are not affected by the state of HGEN.

6.5.6.2 HPCR Host Address Line 8 Enable (HA8EN) Bit 1

If the HA8EN bit is set and the HDI08 is used in multiplexed bus mode, then HA8/HA1 is used as host address line 8 (HA8). If this bit is cleared and the HDI08 is used in multiplexed bus mode, then HA8/HA1 is configured as GPIO pin according to the value of HDDR and HDR registers. HA8EN is ignored when the HDI08 is not in the multiplexed bus mode (HMUX=0).

6.5.6.3 HPCR Host Address Line 9 Enable (HA9EN) Bit 2

If the HA9EN bit is set and the HDI08 is used in multiplexed bus mode, then HA9/HA2 is used as host address line 9 (HA9). If this bit is cleared and the HDI08 is used in multiplexed bus mode, then HA9/HA2 is configured as GPIO pin according to the value of HDDR and HDR registers. HA9EN is ignored when the HDI08 is not in the multiplexed bus mode (HMUX=0).

6.5.6.4 HPCR Host Chip Select Enable (HCSEN) Bit 3

If the HCSEN bit is set, then HCS/HA10 is used as host chip select (HCS) in the non-multiplexed bus mode (HMUX=0), and as host address line 10 (HA10) in the multiplexed bus mode (HMUX=1). If this bit is cleared, then HCS/HA10 is configured as GPIO pin according to the value of HDDR and HDR registers.

6.5.6.5 HPCR Host Request Enable (HREN) Bit 4

The HREN bit controls the host request signals. If HREN is set and the HDI08 is in the single host request mode (HDRQ=0 in the ICR), HOREQ/HTRQ is configured as the host request (HOREQ) output.

If HREN is set in the double host request mode (HDRQ=1 in the ICR), HOREQ/HTRQ is configured as the host transmit request (HTRQ) output and HACK/HRRQ as the host receive request (HRRQ) output.

If HREN is cleared, HOREQ/HTRQ and HACK/HRRQ are configured as GPIO pins according to the value of HDDR and HDR registers.

6.5.6.6 HPCR Host Acknowledge Enable (HAEN) Bit 5

The HAEN bit controls the HACK signal. In the single host request mode (HDRQ=0 in the ICR), if HAEN and HREN are both set, HACK/HRRQ is configured as the host acknowledge (HACK) input. If HAEN or HREN is cleared, HACK/HRRQ is configured as a GPIO pin according to the value of HDDR and HDR registers. In the double host request mode (HDRQ=1 in the ICR), HAEN is ignored.

6.5.6.7 HPCR Host Enable (HEN) Bit 6

If the HEN bit is set, the HDI08 operation is enabled as Host Interface. If cleared, the HDI08 is not active, and all the HDI08 pins are configured as GPIO pins according to the value of HDDR and HDR registers.

6.5.6.8 HPCR Reserved Bit 7

This bit is reserved. It reads as zero and should be written with zero for future compatibility.

6.5.6.9 HPCR Host Request Open Drain (HROD) Bit 8

The HROD bit controls the output drive of the host request signals. In the single host request mode (HDRQ=0 in ICR), if HROD is cleared and host requests are enabled (HREN=1 and HEN=1 in HPCR), the HOREQ signal is always driven. If HROD is set and host requests are enabled, the HOREQ signal is an open drain output.

In the double host request mode (HDRQ=1 in the ICR), if HROD is cleared and host requests are enabled (HREN=1 and HEN=1 in the HPCR), the HTRQ and HRRQ signals are always driven. If HROD is set and host requests are enabled, the HTRQ and HRRQ signals are open drain outputs.

6.5.6.10 HPCR Host Data Strobe Polarity (HDSP) Bit 9

If the HDSP bit is cleared, the data strobe signals are configured as active low inputs, and data is transferred when the data strobe is low. If HDSP is set, the data strobe signals are configured as active high inputs, and data is transferred when the data strobe is high. The data strobe signals are either HDS by itself or HRD and HWR together.

6.5.6.11 HPCR Host Address Strobe Polarity (HASP) Bit 10

If the HASP bit is cleared, the address strobe ($\overline{\text{HAS}}$) signal is an active low input, and the address on the host address/data bus is sampled when the $\overline{\text{HAS}}$ signal is low. If HASP is set, HAS is an active high address strobe input, and the address on the host address/data bus 8 is sampled when the HAS signal is high.

6.5.6.12 HPCR Host Multiplexed bus (HMUX) Bit 11

If the HMUX bit is set, the HDI08 latches the lower portion of a multiplexed address/data bus. In this mode the internal address line values of the host registers are taken from the internal latch. If HMUX is cleared, it indicates that the HDI08 is connected to a non-multiplexed type of bus, and the address lines are taken from the HDI08 input signals.

6.5.6.13 HPCR Host Dual Data Strobe (HDDS) Bit 12

If the HDDS bit is cleared, the HDI08 operates in the single strobe bus mode. In this mode, the bus has a single data strobe signal for both reads and writes. If HDDS is set, the HDI08 operates in the dual strobe bus mode. In this mode, the bus has two separate data strobes, one for data reads, the other for data writes. See Figure 6-7 and Figure 6-8 for more information on the two types of buses.

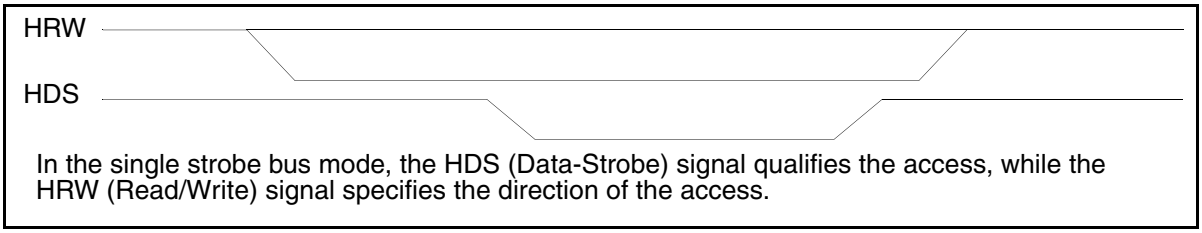

HRW
HDS
In the single strobe bus mode, the HDS (Data-Strobe) signal qualifies the access, while the HRW (Read/Write) signal specifies the direction of the access.

Figure 6-7 Single strobe bus

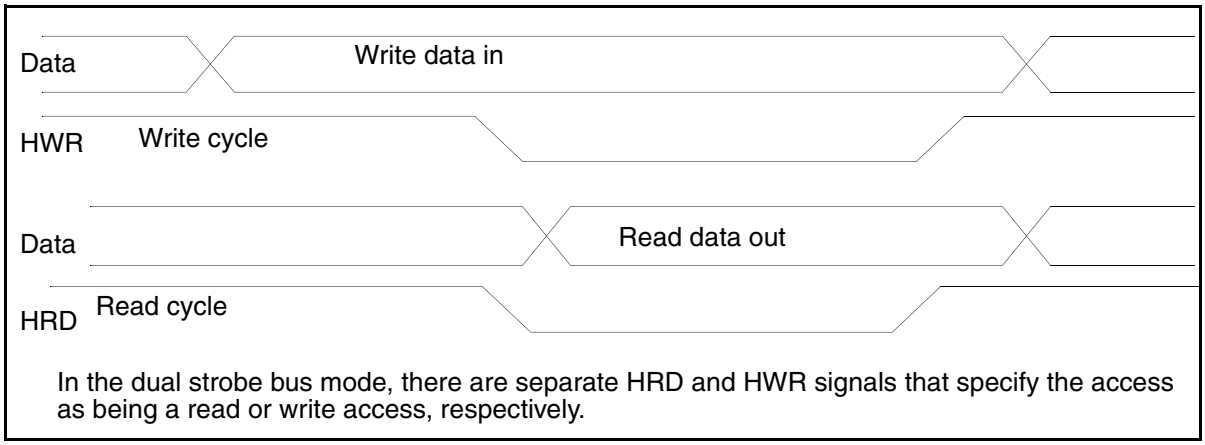

Data
Write data in
HWR
Write cycle
Data
Read data out
HRD
Read cycle
In the dual strobe bus mode, there are separate HRD and HWR signals that specify the access as being a read or write access, respectively.

Figure 6-8 Dual strobes bus

6.5.6.14 HPCR Host Chip Select Polarity (HCSP) Bit 13

If the HCSP bit is cleared, the chip select ($\overline{\text{HCS}}$) signal is configured as an active low input and the HDI08 is selected when the $\overline{\text{HCS}}$ signal is low. If HCSP is set, HCS is configured as an active high input and the HDI08 is selected when the HCS signal is high. This bit is ignored in the multiplexed mode.

6.5.6.15 HPCR Host Request Polarity (HRP) Bit 14

The HRP bit controls the polarity of the host request signals. In the single host request mode (HDRQ=0 in the ICR), if HRP is cleared and host requests are enabled (HREN=1 and HEN=1 in the HPCR), the $\overline{\text{HOREQ}}$ signal is an active low output. If HRP is set and host requests are enabled, the HOREQ signal is an active high output.

In the double host request mode (HDRQ=1 in the ICR), if HRP is cleared and host requests are enabled (HREN=1 and HEN=1 in the HPCR), the $\overline{\text{HTRQ}}$ and $\overline{\text{HRRQ}}$ signals are active low outputs. If HRP is set and host requests are enabled, the HTRQ and HRRQ signals are active high outputs.

6.5.6.16 HPCR Host Acknowledge Polarity (HAP) Bit 15

If the HAP bit is cleared, the host acknowledge ($\overline{\text{HACK}}$) signal is configured as an active low input, and the HDI08 drives the contents of the HIVR register onto the host bus when the $\overline{\text{HACK}}$ signal is low. If HAP is set, HACK is configured as an active high input, and the HDI08 outputs the contents of the HIVR register when the HACK signal is high.

6.5.7 Data direction register (HDDR)

The HDDR controls the direction of the data flow for each of the HDI08 pins configured as GPIO. Even when the HDI08 is used as the host interface, some of its unused signals may be configured as GPIO pins. For information on the HDI08 GPIO configuration options, see Section 6.6.8, "General Purpose INPUT/OUTPUT (GPIO)". If bit DRxx is set, the corresponding HDI08 pin is configured as an output signal. If bit DRxx is cleared, the corresponding HDI08 pin is configured as an input signal. See Table 6-6.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR15	DR14	DR13	DR12	DR11	DR10	DR9	DR8	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0

Figure 6-9 Host Data Direction Register (HDDR) (X:$FFFFC8)

6.5.8 Host Data Register (HDR)

The HDR register holds the data value of the corresponding bits of the HDI08 pins which are configured as GPIO pins. The functionality of the Dxx bit depends on the corresponding HDDR bit (DRxx). See Table 6-6.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Figure 6-10 Host Data Register (HDR) (X:$FFFFC9)

Table 6-6 HDR and HDDR Functionality

HDDR	HDR	
DRxx	Dxx	
	GPIO pin[1]	non-GPIO pin[1]
0	Read only bit. The value read is the binary value of the pin. The corresponding pin is configured as an input.	Read only bit. Does not contain significant data.
1	Read/write bit. The value written is the value read. The corresponding pin is configured as an output, and is driven with the data written to Dxx.	Read/write bit. The value written is the value read.

[1] Defined by the selected configuration

6.5.9 DSP-Side Registers After Reset

Table 6-7 shows the results of the four reset types on the bits in each of the HDI08 registers accessible by the DSP core. The hardware reset (HW) is caused by the $\overline{\text{RESET}}$ signal. The software reset (SW) is caused by executing the RESET instruction. The individual reset (IR) is caused by clearing the HEN bit (HPCR bit 6). The stop reset (ST) is caused by executing the STOP instruction.

Table 6-7 DSP-Side Registers after Reset

Register Name	Register Data	Reset Type			
		HW Reset	SW Reset	IR Reset	ST Reset
HCR	All bits	0	0	—	—
HPCR	All bits	0	0	—	—
HSR	HF[1:0]	0	0	—	—
	HCP	0	0	0	0
	HTDE	1	1	1	1
	HRDF	0	0	0	0
	DMA	0	0	—	—
HBAR	BA[10:3]	$80	$80	—	—
HDDR	DR[15:0]	0	0	—	—
HDR	D[15:0]	—	—	—	—
HORX	HORX[23:0]	empty	empty	empty	empty
HOTX	HOTX[23:0]	empty	empty	empty	empty

Note: A long dash (—) denotes that the register value is not affected by the specified reset.

6.5.10 Host Interface DSP Core Interrupts

The HDI08 may request interrupt service from either the DSP core or the host processor. The DSP core interrupts are internal and do not require the use of an external interrupt pin. When the appropriate interrupt enable bit in the HCR is set, an interrupt condition caused by the host processor sets the appropriate bit in the HSR, generating an interrupt request to the DSP core. The DSP core acknowledges interrupts caused by the host processor by jumping to the appropriate interrupt service routine. The three possible interrupts are as follows:

- Host command
- Transmit data register empty
- Receive data register full

Although there is a set of vectors reserved for host command use, the host command can access any interrupt vector in the interrupt vector table. The DSP interrupt service routine must read or write the appropriate HDI08 register (clearing HRDF or HTDE, for example) to clear the interrupt. In the case of host command interrupts, the interrupt acknowledge from the DSP core program controller clears the pending interrupt condition. Figure 6-11 illustrates the HSR-HCR operation.

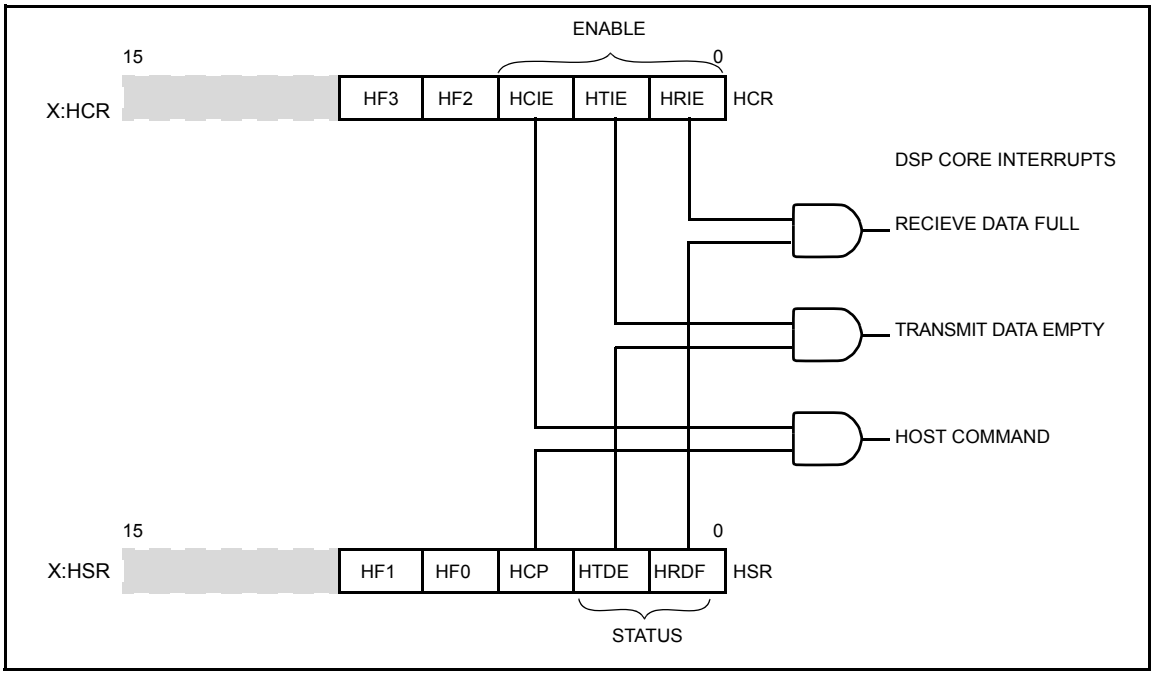


Figure 6-11 HSR-HCR Operation

6.6 HDI08 – External Host Programmer's Model

The HDI08 has been designed to provide a simple, high speed interface to a host processor. To the host bus, the HDI08 appears to be eight byte-wide registers. Separate transmit and receive data registers are double-buffered to allow the DSP core and host processor to transfer data efficiently at high speed. The host may access the HDI08 asynchronously by using polling techniques or interrupt-based techniques.

The HDI08 appears to the host processor as a memory-mapped peripheral occupying 8 bytes in the host processor address space (See Table 6-8). The eight HDI08 include the following:

- A control register (ICR)
- A status register (ISR)
- Three data registers (RXH/TXH, RXM/TXM and RXL/TXL)
- Two vector registers (IVR and CVR)

These registers can be accessed only by the host processor.

Host processors may use standard host processor instructions (e.g., byte move) and addressing modes to communicate with the HDI08 registers. The HDI08 registers are aligned so that 8-bit host processors can use 8/16/24-bit load and store instructions for data transfers. The HOREQ/HTRQ and HACK/HRRQ handshake flags are provided for polled or interrupt-driven data transfers with the host processor. Because the DSP interrupt response, most host microprocessors can load or store data at their maximum programmed I/O instruction rate without testing the handshake flags for each transfer. If full handshake is not needed, the host processor can treat the DSP as a fast device, and data can be transferred between the host processor and the DSP at the fastest host processor data rate.

One of the most innovative features of the host interface is the host command feature. With this feature, the host processor can issue vectored interrupt requests to the DSP core. The host may select any of 128 DSP interrupt routines to be executed by writing a vector address register in the HDI08. This flexibility allows the host programmer to execute up to 128 pre-programmed functions inside the DSP. For example, host interrupts can allow the host processor to read or write DSP registers (X, Y, or program memory locations), force interrupt handlers (e.g. $\overline{\text{IRQA}}$, $\overline{\text{IRQB}}$, etc. interrupt routines), and perform control and debugging operations if interrupt routines are implemented in the DSP to perform these tasks.

Table 6-8 HDI08 Host Side Register Map

Host Address	Big Endian HLEND=0		Little Endian HLEND=1	Function
0	ICR		ICR	Interface Control
1	CVR		CVR	Command Vector
2	ISR		ISR	Interface Status
3	IVR		IVR	Interrupt Vector
4	00000000		00000000	Unused
5	RXH/TXH[1]		RXL/TXL	Receive/Transmit Bytes
6	RXM/TXM		RXM/TXM	
7	RXL/TXL		RXH/TXH	
	↕		↕	
	Host Data Bus H0 - H7		Host Data Bus H0 - H7	

[1] The RXH/TXH register is always mapped to the most significant byte of the DSP word.

6.6.1 Interface Control Register (ICR)

The ICR is an 8-bit read/write control register used by the host processor to control the HDI08 interrupts and flags. The ICR cannot be accessed by the DSP core. The ICR is a read/write register, which allows the use of bit manipulation instructions on control register bits. The control bits are described in the following paragraphs.

Bits 2, 5 and 6 of the ICR are affected by the condition of HDM[2:0] (HCR bits 5-7), as shown in Figure 6-12.

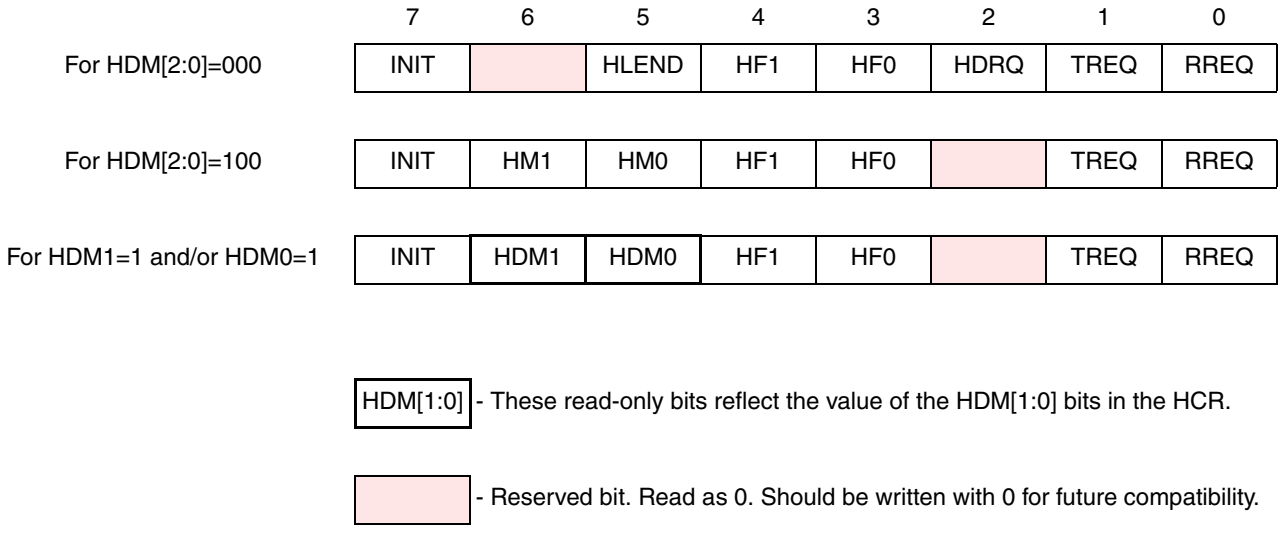

7 6 5 4 3 2 1 0
For HDM[2:0]=000
INIT HLEND HF1 HF0 HDRQ TREQ RREQ
For HDM[2:0]=100
INIT HM1 HM0 HF1 HF0 TREQ RREQ
For HDM1=1 and/or HDM0=1
INIT HDM1 HDM0 HF1 HF0 TREQ RREQ
HDM[1:0] - These read-only bits reflect the value of the HDM[1:0] bits in the HCR.
- Reserved bit. Read as 0. Should be written with 0 for future compatibility.

Figure 6-12 Interface Control Register (ICR)

6.6.1.1 ICR Receive Request Enable (RREQ) Bit 0

In interrupt mode (HDM[2:0]=000 or HM[1:0]=00), RREQ is used to enable host receive data requests via the host request (HOREQ or HRRQ) signal when the receive data register full (RXDF) status bit in the ISR is set. If RREQ is cleared, RXDF requests are disabled. If RREQ is set, the host request signal (HOREQ or HRRQ) is asserted if RXDF is set.

In the DMA modes where HDM[2:0]=100 and (HM1≠0 or HM0≠0), RREQ must be set and TREQ must be cleared to direct DMA transfers from DSP to host. In the other DMA modes, RREQ is ignored.

Table 6-9 summarizes the effect of RREQ and TREQ on the HOREQ, HTRQ and HRRQ signals.

6.6.1.2 ICR Transmit Request Enable (TREQ) Bit 1

In interrupt mode (HDM[2:0]=000 or HM[1:0]=00), TREQ is used to enable host transmit data requests via the host request (HOREQ or HTRQ) signal when the transmit data register empty (TXDE) status bit in the ISR is set. If TREQ is cleared, TXDE requests are disabled. If TREQ is set, the host request signal (HOREQ or HTRQ) is asserted if TXDE is set.

In the DMA modes where HDM[2:0]=100 and (HM1≠0 or HM0≠0), TREQ must be set and RREQ must be cleared to direct DMA transfers from host to DSP. In the other DMA modes, TREQ is ignored.

Table 6-9 summarizes the effect of RREQ and TREQ on the HOREQ, HTRQ and HRRQ signals.

Table 6-9 TREQ RREQ Interrupt Mode (HDM[2:0]=000 or HM[1:0]=00)

TREQ	RREQ	HDRQ=0	HDRQ=1	
		HOREQ signal	HTRQ signal	HRRQ signal
0	0	No Interrupts (Polling)	No Interrupts (Polling)	No Interrupts (Polling)
0	1	RXDF Request (Interrupt)	No Interrupts (Polling)	RXDF Request (Interrupt)
1	0	TXDE Request (Interrupt)	TXDE Request (Interrupt)	No Interrupts (Polling)
1	1	RXDF and TXDE Requests (Interrupts)	TXDE Request (Interrupt)	RXDF Request (Interrupt)

Table 6-10 TREQ RREQ DMA Mode (HM1≠0 or HM0≠0)

TREQ	RREQ	HDRQ=0	HDRQ=1	
		HOREQ signal	HTRQ signal	HRRQ signal
0	0	No DMA request	No DMA request	No DMA request
0	1	DSP to Host Request (RX)	No DMA request	DSP to Host Request (RX)
1	0	Host to DSP Request (TX)	Host to DSP Request (TX)	No DMA request
1	1	Reserved	Reserved	Reserved

6.6.1.3 ICR Double Host Request (HDRQ) Bit 2

The HDRQ bit determines the functions of the HOREQ/HTRQ and HACK/HRRQ signals as shown in Table 6-11.

Table 6-11 HDRQ

HDRQ	HOREQ/HTRQ pin	HACK/HRRQ pin
0	HOREQ signal	HACK signal
1	HTRQ signal	HRRQ signal

6.6.1.4 ICR Host Flag 0 (HF0) Bit 3

The HF0 bit is used as a general purpose flag for host-to-DSP communication. HF0 may be set or cleared by the host processor and cannot be changed by the DSP core. HF0 is reflected in the HSR on the DSP side of the HDI08.

6.6.1.5 ICR Host Flag 1 (HF1) Bit 4

The HF1 bit is used as a general purpose flag for host-to-DSP communication. HF1 may be set or cleared by the host processor and cannot be changed by the DSP core. HF1 is reflected in the HSR on the DSP side of the HDI08.

6.6.1.6 ICR Host Little Endian (HLEND) Bit 5

If the HLEND bit is cleared, the HDI08 can be accessed by the host in big endian byte order. If set, the HDI08 can be accessed by the host in little endian byte order. If the HLEND bit is cleared, the RXH/TXH register is located at address $5, the RXM/TXM register is located at address $6, and the RXL/TXL register is located at address $7. If the HLEND bit is set, the RXH/TXH register is located at address $7, the RXM/TXM register is located at address $6, and the RXL/TXL is located at address $5. See Table 6-8 for an illustration of the effect of HLEND.

The HLEND function is available only if HDM[2:0]=000 in the host control register (HCR). When HLEND is available, the ICR bit 6 has no function and should be regarded as reserved.

Bits 6 and 5 function as read/write HM[1:0] bits only when the HCR bits HDM[2:0]=100 (See Table 6-5). The HM0 and HM1 bits select the transfer mode of the HDI08, as shown in Table 6-12.

Table 6-12 Host Mode Bit Definition

HM1	HM0	Mode
0	0	Interrupt Mode (DMA Off)
0	1	DMA Mode (24 Bit)
1	0	DMA Mode (16 Bit)
1	1	DMA Mode (8 Bit)

When both HM1 and HM0 are cleared, the DMA mode is disabled and the interrupt mode is enabled. In interrupt mode, the TREQ and RREQ control bits are used for host processor interrupt control via the external HOREQ output signal, and the HACK input signal is used for the MC68000 Family vectored interrupt acknowledge input.

When HM1 and/or HM0 are set, they enable the DMA mode and determine the size of the DMA word to be transferred. In the DMA mode, the HOREQ signal is used to request DMA transfers, the TREQ and RREQ bits select the direction of DMA transfers (see Table 6-10), and the HACK input signal is used as a DMA transfer acknowledge input. If the DMA direction is from DSP to host, the contents of the selected register are enabled onto the host data bus when HACK is asserted. If the DMA direction is from host to DSP, the selected register is written from the host data bus when HACK is asserted.

The size of the DMA word to be transferred is determined by the DMA control bits, HM0 and HM1. The HDI08 host side data register selected during a DMA transfer is determined by a 2-bit address counter, which is preloaded with the value in HM1 and HM0. The address counter substitutes for the HA1 and HA0 host address signals of the HDI08 during a DMA transfer. The host address signal HA2 is forced to one during each DMA transfer. The address counter can be initialized with the INIT bit feature. After each DMA transfer on the host data bus, the address counter is incremented to the next data register. When the address counter reaches the highest register (RXL or TXL), the address counter is not incremented but is loaded with the value in HM1 and HM0. This allows 8-, 16- or 24-bit data to be transferred in a circular fashion and eliminates the need for the DMA controller to supply the HA2, HA1, and HA0 address signals. For 16- or 24-bit data transfers, the DSP CPU interrupt rate is reduced by a factor of 2 or 3, respectively,

from the host request rate – i.e., for every two or three host processor data transfers of one byte each, there is only one 24-bit DSP CPU interrupt.

If either HDM1 or HDM0 in the HCR register are set, bits 6 and 5 become read-only bits that reflect the value of HDM[1:0].

6.6.1.7 ICR Initialize Bit (INIT) Bit 7

The INIT bit is used by the host processor to force initialization of the HDI08 hardware. During initialization, the HDI08 transmit and receive control bits are configured.

Using the INIT bit to initialize the HDI08 hardware may or may not be necessary, depending on the software design of the interface.

The type of initialization done when the INIT bit is set depends on the state of TREQ and RREQ in the HDI08. The INIT command, which is local to the HDI08, is designed to conveniently configure the HDI08 into the desired data transfer mode. The effect of the INIT command is described in Table 6-13. When the host sets the INIT bit, the HDI08 hardware executes the INIT command. The interface hardware clears the INIT bit after the command has been executed.

Table 6-13 INIT Command Effect

TREQ	RREQ	After INIT Execution	Transfer Direction Initialized
0	0	INIT=0	None
0	1	INIT=0; RXDF=0; HTDE=1	DSP to Host
1	0	INIT=0; TXDE=1; HRDF=0	Host to DSP
1	1	INIT=0; RXDF=0; HTDE=1; TXDE=1; HRDF=0	Host to/from DSP

6.6.2 Command Vector Register (CVR)

The CVR is used by the host processor to cause the DSP core to execute an interrupt. The host command feature is independent of any of the data transfer mechanisms in the HDI08. It can be used to invoke execution of any of the 128 possible interrupt routines in the DSP core.

7	6	5	4	3	2	1	0
HC	HV6	HV5	HV4	HV3	HV2	HV1	HV0

Figure 6-13 Command Vector Register (CVR)

6.6.2.1 CVR Host Vector (HV[6:0]) Bits 0–6

The seven HV bits select the host command interrupt address to be used by the host command interrupt logic. When the host command interrupt is recognized by the DSP interrupt control logic, the address of the interrupt routine taken is 2 ∗ HV. The host can write HC and HV in the same write cycle.

The host processor can select the starting address of any of the 128 possible interrupt routines in the DSP by writing the interrupt routine address divided by 2 into the HV bits. The host processor can thus force execution of any of the existing interrupt handlers ($\overline{\text{IRQA}}$, $\overline{\text{IRQB}}$, etc.) and can use any of the reserved or otherwise unused addresses provided they have been pre-programmed in the DSP. HV[6:0] is set to $32 (vector location $0064) by hardware, software, individual and stop resets.

6.6.2.2 CVR Host Command Bit (HC) Bit 7

The HC bit is used by the host processor to handshake the execution of host command interrupts. Normally, the host processor sets HC to request the host command interrupt from the DSP core. When the host command interrupt is acknowledged by the DSP core, the HC bit is cleared by the HDI08 hardware. The host processor can read the state of HC to determine when the host command has been accepted. After setting HC, the host must not write to the CVR again until HC is cleared by the HDI08 hardware. Setting HC causes the host command pending (HCP) in the HSR to be set. The host can write to the HC and HV bits in the same write cycle.

6.6.3 Interface Status Register (ISR)

The ISR is an 8-bit read-only status register used by the host processor to interrogate the status and flags of the HDI08. The host processor can write to this address without affecting the internal state of the HDI08, which is useful if the user desires to access all of the HDI08 registers by stepping through the HDI08 addresses. The ISR cannot be accessed by the DSP core. The ISR bits are described in the following paragraphs.

7	6	5	4	3	2	1	0
HREQ			HF3	HF2	TRDY	TXDE	RXDF

- Reserved bit. Read as 0. Should be written with 0 for future compatibility.

Figure 6-14 Interface Status Register (ISR)

6.6.3.1 ISR Receive Data Register Full (RXDF) Bit 0

The RXDF bit indicates that the receive byte registers (RXH:RXM:RXL) contain data from the DSP core and may be read by the host processor. RXDF is set when the contents of HOTX is transferred to the receive byte registers. RXDF is cleared when the receive data (RXL or RXH according to HLEND bit) register is read by the host processor. RXDF can be cleared by the host processor using the initialize function. RXDF may be used to assert the external HOREQ signal if the RREQ bit is set. Regardless of whether the RXDF interrupt is enabled, RXDF indicates whether the RX registers are full and data can be latched out (so that polling techniques may be used by the host processor).

6.6.3.2 ISR Transmit Data Register Empty (TXDE) Bit 1

The TXDE bit indicates that the transmit byte registers (TXH:TXM:TXL) are empty and can be written by the host processor. TXDE is set when the contents of the transmit byte registers are transferred to the HORX register. TXDE is cleared when the transmit (TXL or TXH according to HLEND bit) register is

written by the host processor. TXDE can be set by the host processor using the initialize feature. TXDE may be used to assert the external HOREQ signal if the TREQ bit is set. Regardless of whether the TXDE interrupt is enabled, TXDE indicates whether the TX registers are full and data can be latched in (so that polling techniques may be used by the host processor).

6.6.3.3 ISR Transmitter Ready (TRDY) Bit 2

The TRDY status bit indicates that TXH:TXM:TXL and the HORX registers are empty.

$$TRDY = TXDE \bullet \overline{HRDF}$$

If TRDY is set, the data that the host processor writes to TXH:TXM:TXL is immediately transferred to the DSP side of the HDI08. This feature has many applications. For example, if the host processor issues a host command which causes the DSP core to read the HORX, the host processor can be guaranteed that the data it just transferred to the HDI08 is what is being received by the DSP core.

6.6.3.4 ISR Host Flag 2 (HF2) Bit 3

The HF2 bit in the ISR indicates the state of host flag 2 in the HCR on the DSP side. HF2 can be changed only by the DSP (see Section 6.5.3.4, "HCR Host Flags 2,3 (HF2,HF3) Bits 3-4").

6.6.3.5 ISR Host Flag 3 (HF3) Bit 4

The HF3 bit in the ISR indicates the state of host flag 3 in the HCR on the DSP side. HF3 can be changed only by the DSP (see Section 6.5.3.4, "HCR Host Flags 2,3 (HF2,HF3) Bits 3-4").

6.6.3.6 ISR Reserved Bits 5-6

These bits are reserved. They read as zero and should be written with zero for future compatibility.

6.6.3.7 ISR Host Request (HREQ) Bit 7

The HREQ bit indicates the status of the external host request output signal (HOREQ) if HDRQ is cleared. If HDRQ is set, it indicates the status of the external transmit and receive request output signals (HTRQ and HRRQ).

Table 6-14 Host Request Status (HREQ)

HREQ	Status [HDRQ=0]	Status [HDRQ=1]
0	HOREQ deasserted; no host processor interrupt is requested	HTRQ and HRRQ deasserted; no host processor interrupts are requested
1	HOREQ asserted; a host processor interrupt is requested	HTRQ and/or HRRQ asserted; host processor interrupts are requested

The HREQ bit may be set from either or both of two conditions – either the receive byte registers are full or the transmit byte registers are empty. These conditions are indicated by the ISR RXDF and TXDE status bits, respectively. If the interrupt source has been enabled by the associated request enable bit in the ICR, HREQ is set if one or more of the two enabled interrupt sources is set.

6.6.4 Interrupt Vector Register (IVR)

The IVR is an 8-bit read/write register which typically contains the interrupt vector number used with MC68000 Family processor vectored interrupts. Only the host processor can read and write this register. The contents of IVR are placed on the host data bus (H0–H7) when both the HOREQ and HACK signals are asserted. The contents of this register are initialized to $0F by hardware or software reset, which corresponds to the uninitialized interrupt vector in the MC68000 Family.

7	6	5	4	3	2	1	0
IV7	IV6	IV5	IV4	IV3	IV2	IV1	IV0

Figure 6-15 Interrupt Vector Register (IVR)

6.6.5 Receive Byte Registers (RXH:RXM:RXL)

The receive byte registers are viewed by the host processor as three 8-bit read-only registers. These registers are the receive high register (RXH), the receive middle register (RXM) and the receive low register (RXL). They receive data from the high, middle and low bytes, respectively, of the HOTX register and are selected by the external host address inputs (HA2, HA1 and HA0) during a host processor read operation.

The memory locations of the receive byte registers are determined by the HLEND bit in the ICR. If the HLEND bit is set, the RXH is located at address $7, RXM at $6 and RXL at $5. If the HLEND bit is cleared, the RXH is located at address $5, RXM at $6 and RXL at $7.

When data is transferred from the HOTX register to the receive byte registers, the receive data register full (RXDF) bit is set. The host processor may program the RREQ bit to assert the external HOREQ/HRRQ signal when RXDF is set. This indicates that the HDI08 has a full word (either 8, 16 or 24 bits) for the host processor. When the host reads the receive byte register at host address $7, the RXDF bit is cleared.

6.6.6 Transmit Byte Registers (TXH:TXM:TXL)

The transmit byte registers are viewed as three 8-bit write-only registers by the host processor. These registers are the transmit high register (TXH), the transmit middle register (TXM) and the transmit low register (TXL). These registers send data to the high, middle and low bytes, respectively, of the HORX register and are selected by the external host address inputs (HA2, HA1 and HA0) during a host processor write operation.

If the HLEND bit in the ICR is cleared, the TXH is located at address $5, TXM at $6 and TXL at $7. If the HLEND bit in the ICR is set, the TXH is located at address $7, TXM at $6 and TXL at $5.

Data may be written into the transmit byte registers when the transmit data register empty (TXDE) bit is set. The host processor may program the TREQ bit to assert the external HOREQ/HTRQ signal when TXDE is set. This informs the host processor that the transmit byte registers are empty. Writing to the data register at host address $7 clears the TXDE bit. The contents of the transmit byte registers are transferred as 24-bit data to the HORX register when both TXDE and the HRDF bit are cleared. This transfer operation sets TXDE and HRDF.

6.6.7 Host Side Registers After Reset

Table 6-15 shows the result of the four kinds of reset on bits in each of the HDI08 registers seen by the host processor. The hardware reset (HW) is caused by asserting the $\overline{\text{RESET}}$ signal. The software reset (SW) is caused by executing the RESET instruction. The individual reset (IR) is caused by clearing the HEN bit in the HPCR register. The stop reset (ST) is caused by executing the STOP instruction.

Table 6-15 Host Side Registers After Reset

Register Name	Register Data	Reset Type			
		HW Reset	SW Reset	IR Reset	ST Reset
ICR	All Bits	0	0	—	—
CVR	HC	0	0	0	0
	HV[6:0]	$32	$32	—	—
ISR	HREQ	0	0	1 if TREQ is set; 0 otherwise	1 if TREQ is set; 0 otherwise
	HF3-HF2	0	0	—	—
	TRDY	1	1	1	1
	TXDE	1	1	1	1
	RXDF	0	0	0	0
IVR	IV[7:0]	$0F	$0F	—	—
RX	RXH:RXM:RXL	empty	empty	empty	empty
TX	TXH:TXM:TXL	empty	empty	empty	empty

Note: A long dash (—) denotes that the register value is not affected by the specified reset.

6.6.8 General Purpose INPUT/OUTPUT (GPIO)

When configured as general-purpose I/O, the HDI08 is viewed by the DSP core as memory-mapped registers (see Section 6.5, "HDI08 – DSP-Side Programmer's Model") that control up to 16 I/O pins. The software and hardware resets clear all DSP-side control registers and configure the HDI08 as GPIO with all 16 signals disconnected. External circuitry connected to the HDI08 may need external pull-up/pull-down resistors until the signals are configured for operation. The registers cleared are the HPCR, HDDR and HDR. Selection between GPIO and HDI08 is made by clearing HPCR bits 6 through 1 for GPIO or setting these bits for HDI08 functionality. If the HDI08 is in GPIO mode, the HDDR configures each corresponding signal in the HDR as an input signal if the HDDR bit is cleared or as an output signal if the HDDR bit is set (see Section 6.5.7, "Data direction register (HDDR)" and Section 6.5.8, "Host Data Register (HDR)").

6.7 Servicing The Host Interface

The HDI08 can be serviced by using one of the following protocols:

- Polling
- Interrupts

6.7.1 HDI08 Host Processor Data Transfer

To the host processor, the HDI08 appears as a contiguous block of static RAM. To transfer data between itself and the HDI08, the host processor performs the following steps:

1. Asserts the HDI08 address to select the register to be read or written.
2. Selects the direction of the data transfer. If it is writing, the host processor drives the data on the bus.
3. Strobes the data transfer.

6.7.2 Polling

In the polling mode of operation, the HOREQ/HTRQ signal is not connected to the host processor and HACK must be deasserted to ensure IVR data is not being driven on H0-H7 when other registers are being polled.

The host processor first performs a data read transfer to read the ISR register.This allows the host processor to assess the status of the HDI08:

1. If RXDF=1, the receive byte registers are full and therefore a data read can be performed by the host processor.
2. If TXDE=1, the transmit byte registers are empty. A data write can be performed by the host processor.
3. If TRDY=1, the transmit byte registers and the receive data register on the DSP side are empty. Data written by the host processor is transferred directly to the DSP side.
4. If (HF2 • HF3) ≠ 0, depending on how the host flags have been defined, may indicate an application-specific state within the DSP core has been reached. Intervention by the host processor may be required.
5. If HREQ=1, the HOREQ/HTRQ/HRRQ signal has been asserted, and the DSP is requesting the attention of the host processor. One of the previous four conditions exists.

After the appropriate data transfer has been made, the corresponding status bit is updated to reflect the transfer.

If the host processor has issued a command to the DSP by writing the CVR and setting the HC bit, it can read the HC bit in the CVR to determine when the command has been accepted by the interrupt controller in the DSP core. When the command has been accepted for execution, the HC bit is cleared by the interrupt controller in the DSP core.

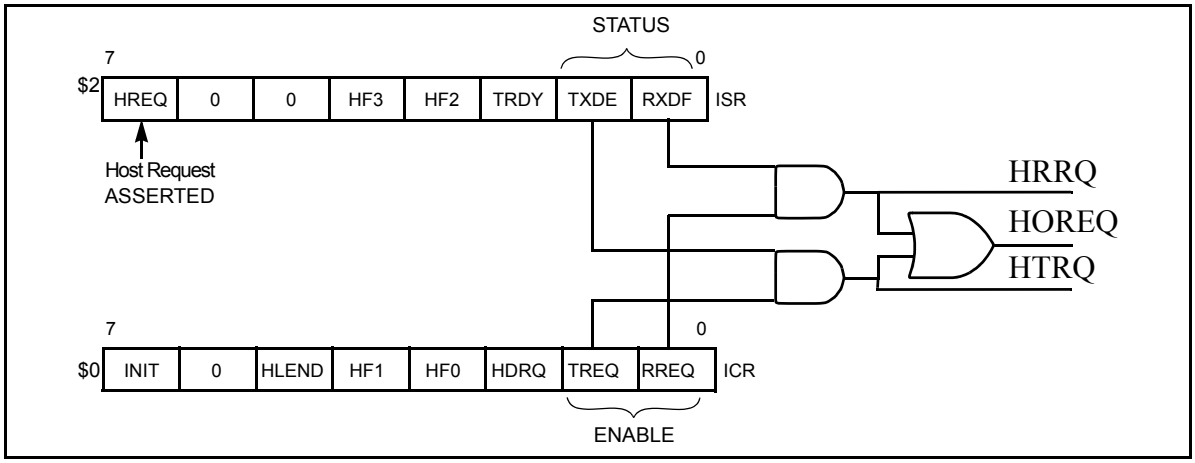


Figure 6-16 HDI08 Host Request Structure

6.7.3 Servicing Interrupts

If either the HOREQ/HTRQ or the HRRQ signal or both are connected to the host processor interrupt inputs, the HDI08 can request service from the host processor by asserting one of these signals. The HOREQ/HTRQ and/or the HRRQ signal is asserted when TXDE=1 and/or RXDF=1 and the corresponding enable bit (TREQ or RREQ, respectively) is set. This is depicted in Figure 6-16.

HOREQ/HTRQ and HRRQ are normally connected to the host processor maskable interrupt inputs. The host processor acknowledges host interrupts by executing an interrupt service routine. The host processor can test RXDF and TXDE to determine the interrupt source. The host processor interrupt service routine must read or write the appropriate HDI08 data register to clear the interrupt. HOREQ/HTRQ and/or HRRQ is deasserted under the following conditions:

- The enabled request is cleared or masked
- The DSP is reset.

If the host processor is a member of the MC68000 family, there is no need for the additional step when the host processor reads the ISR to determine how to respond to an interrupt generated by the DSP. Instead, the DSP automatically sources the contents of the IVR on the data bus when the host processor acknowledges the interrupt by asserting HACK. The contents of the IVR are placed on the host data bus while HOREQ and HACK are simultaneously asserted. The IVR data tells the MC680XX host processor which interrupt routine to execute to service the DSP.

7 Serial Host Interface

7.1 Introduction

The Serial Host Interface (SHI) is a serial I/O interface that provides a path for communication and program/coefficient data transfers between the DSP and an external host processor. The SHI can also communicate with other serial peripheral devices. The SHI supports two well-known and widely used synchronous serial buses: the Freescale Serial Peripheral Interface (SPI) bus and the Philips Inter-Integrated-Circuit Control (I^2C) bus. The SHI supports either bus protocol as either a slave or a single-master device. To minimize DSP overhead, the SHI supports 8-bit, 16-bit and 24-bit data transfers. The SHI has a 1 or 10-word receive FIFO that permits receiving up to 30 bytes before generating a receive interrupt, reducing the overhead for data reception.

When configured in the SPI mode, the SHI can perform the following functions:

- Identify its slave selection (in slave mode)
- Simultaneously transmit (shift out) and receive (shift in) serial data
- Directly operate with 8-, 16- and 24-bit words
- Generate vectored interrupts separately for receive and transmit events and update status bits
- Generate a separate vectored interrupt for a receive exception
- Generate a separate vectored interrupt for a bus-error exception
- Generate the serial clock signal (in master mode)
- Trigger DMA interrupts to service the transmit and receive events

When configured in the I^2C mode, the SHI can perform the following functions:

- Detect/generate start and stop events
- Identify its slave (ID) address (in slave mode)
- Identify the transfer direction (receive/transmit)
- Transfer data byte-wise according to the SCL clock line
- Generate ACK signal following a byte receive
- Inspect ACK signal following a byte transmit
- Directly operate with 8-, 16- and 24-bit words
- Generate vectored interrupts separately for receive and transmit events and update status bits
- Generate a separate vectored interrupt for a receive exception
- Generate a separate vectored interrupt for a bus error exception
- Generate the clock signal (in master mode)
- Trigger DMA interrupts to service the transmit and receive events

7.2 Serial Host Interface Internal Architecture

The DSP views the SHI as a memory-mapped peripheral in the X data memory space. The DSP uses the SHI as a normal memory-mapped peripheral using standard polling or interrupt programming techniques and DMA transfers. Memory mapping allows DSP communication with the SHI registers to be accomplished using standard instructions and addressing modes. In addition, the MOVEP instruction allows interface-to-memory and memory-to-interface data transfers without going through an intermediate register. The DMA controller may be used to service the receive or transmit data path. The single master configuration allows the DSP to directly connect to dumb peripheral devices. For that purpose, a programmable baud-rate generator is included to generate the clock signal for serial transfers. The host side invokes the SHI for communication and data transfer with the DSP through a shift register that may be accessed serially using either the I^2C or the SPI bus protocols. Figure 7-1 shows the SHI block diagram.

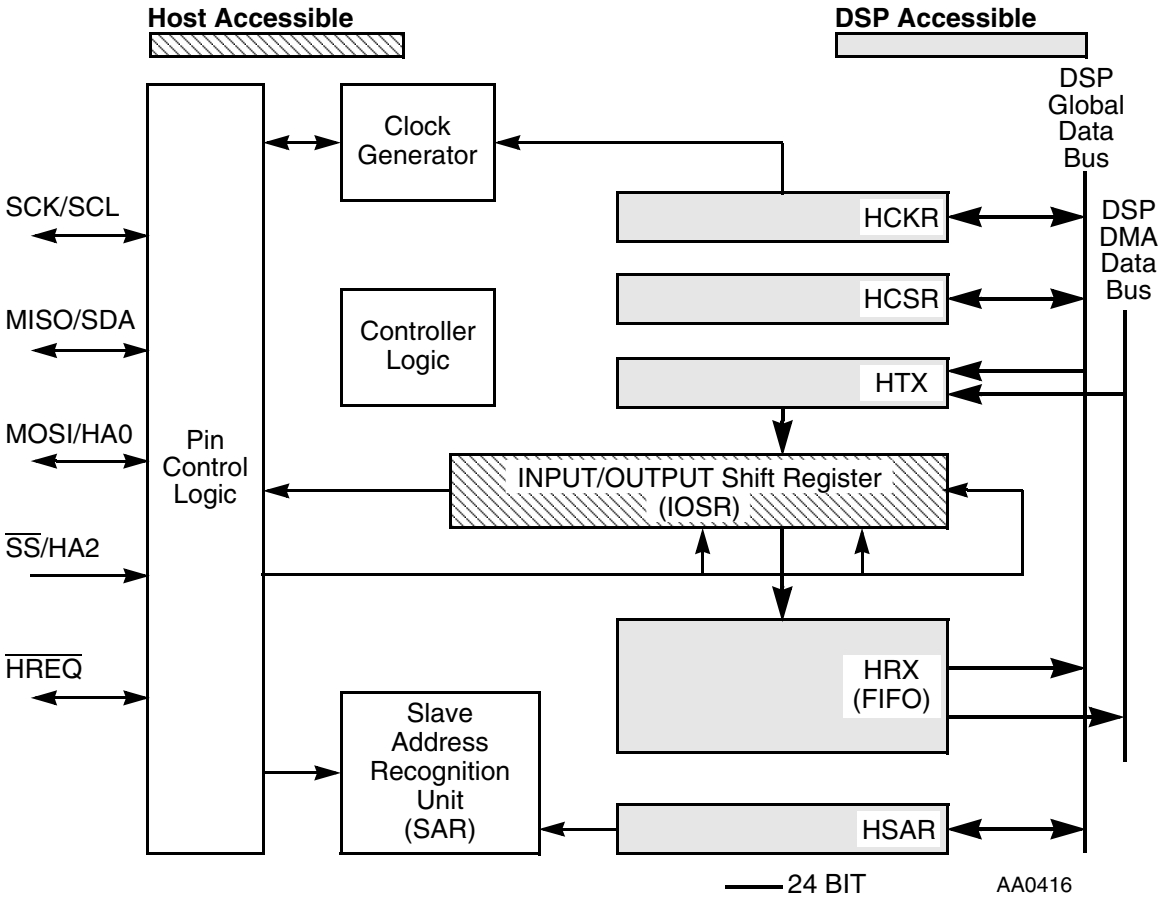


Figure 7-1 Serial Host Interface Block Diagram

7.3 SHI Clock Generator

The SHI clock generator generates the SHI serial clock if the interface operates in the master mode. The clock generator is disabled if the interface operates in the slave mode, except in I^2C mode when the HCKFR bit is set in the HCKR register. When the SHI operates in the slave mode, the clock is external and is input to the SHI (HMST = 0). Figure 7-2 illustrates the internal clock path connections. It is the

user's responsibility to select the proper clock rate within the range as defined in the I^2C and SPI bus specifications.

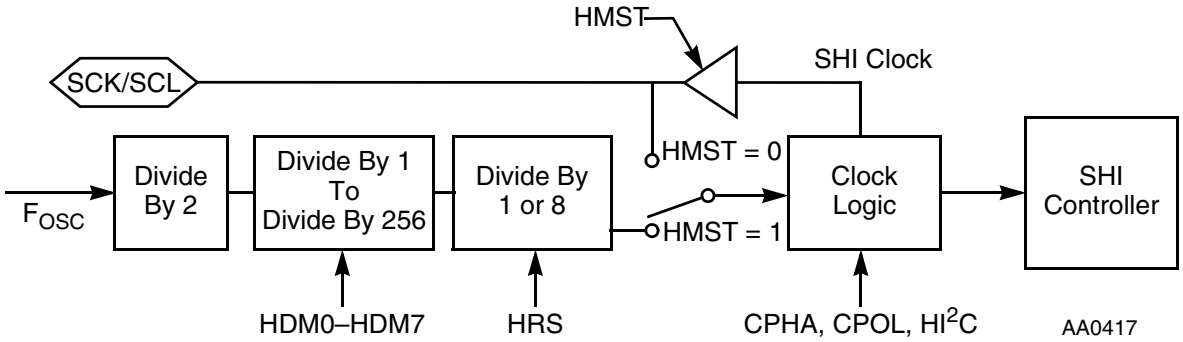


Figure 7-2 SHI Clock Generator

7.4 Serial Host Interface Programming Model

The Serial Host Interface programming model has two parts:

- **Host side**—see Figure 7-3 below and Section 7.4.1, "SHI Input/Output Shift Register (IOSR)—Host Side"
- **DSP side**—see Figure 7-4 and Section 7.4.2, "SHI Host Transmit Data Register (HTX)—DSP Side" through Section 7.4.6, "SHI Control/Status Register (HCSR)—DSP Side" for detailed information.

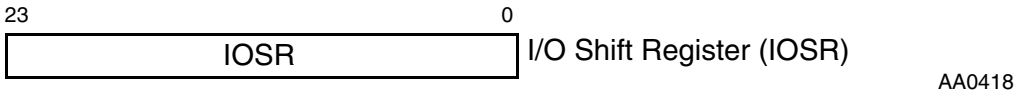


Figure 7-3 SHI Programming Model—Host Side

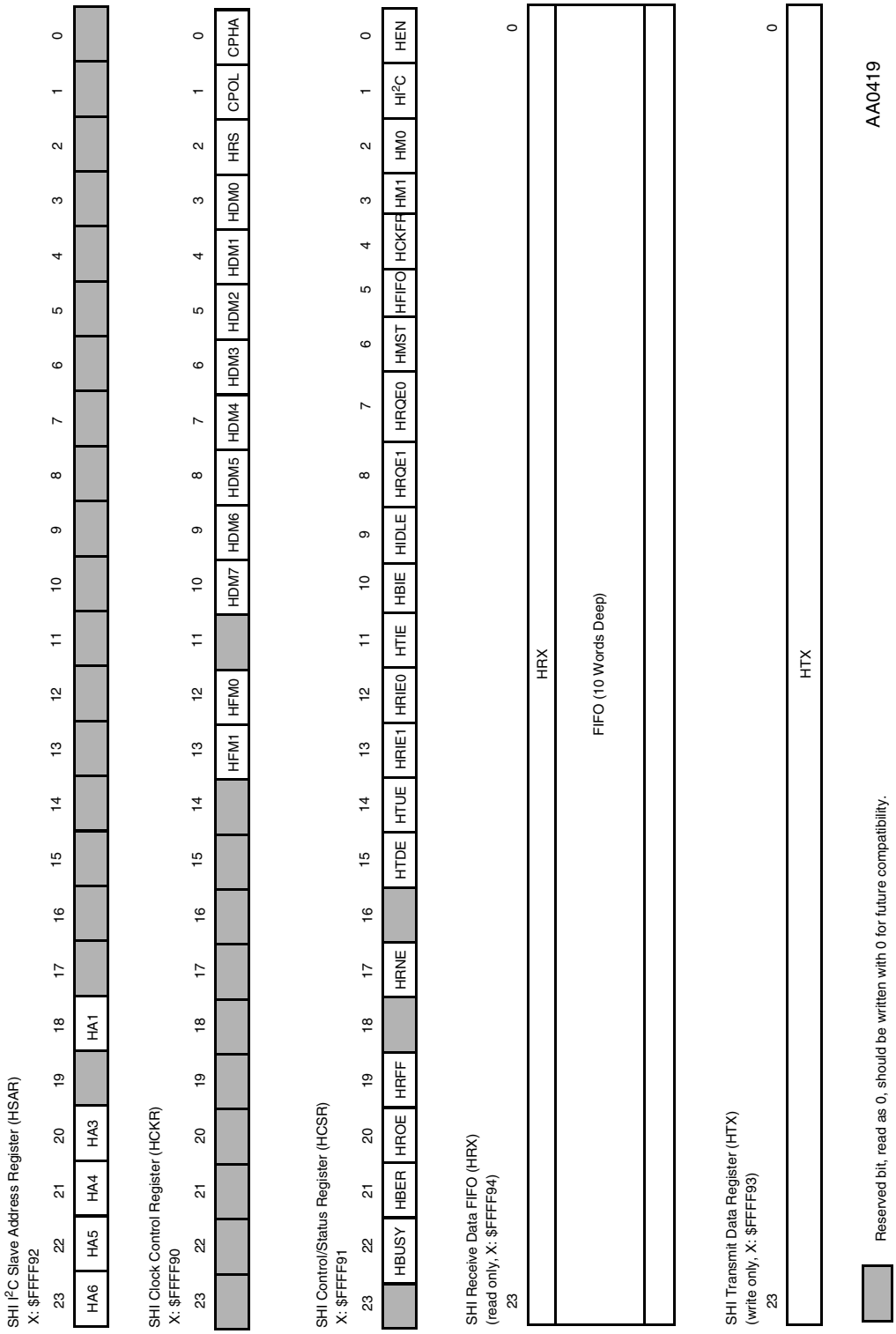

SHI I2C Slave Address Register (HSAR)
X: $FFFF92
HA6 HA5 HA4 HA3 HA1
SHI Clock Control Register (HCKR)
X: $FFFF90
HFM1 HFM0 HDM7 HDM6 HDM5 HDM4 HDM3 HDM2 HDM1 HDM0 HRS CPOL CPHA
SHI Control/Status Register (HCSR)
X: $FFFF91
HBUSY HBER HROE HRFF HRNE HTDE HTUE HRIE1 HRIE0 HTIE HBIE HIDLE HRQE1 HRQE0 HMST HFIFO HCKFR HM1 HM0 HI2C HEN
SHI Receive Data FIFO (HRX)
(read only, X: $FFFF94)
HRX
FIFO (10 Words Deep)
SHI Transmit Data Register (HTX)
(write only, X: $FFFF93)
HTX
Reserved bit, read as 0, should be written with 0 for future compatibility.
AA0419

Figure 7-4 SHI Programming Model—DSP Side

The SHI interrupt vector table is shown in Table 7-1 and the exception priorities generated by the SHI are shown in Table 7-2.

Table 7-1 SHI Interrupt Vectors

Program Address	Interrupt Source
VBA:$0040	SHI Transmit Data
VBA:$0042	SHI Transmit Underrun Error
VBA:$0044	SHI Receive FIFO Not Empty
VBA:$0048	SHI Receive FIFO Full
VBA:$004A	SHI Receive Overrun Error
VBA:$004C	SHI Bus Error

Table 7-2 SHI Internal Interrupt Priorities

Priority	Interrupt
Highest	SHI Bus Error
	SHI Receive Overrun Error
	SHI Transmit Underrun Error
	SHI Receive FIFO Full
	SHI Transmit Data
Lowest	SHI Receive FIFO Not Empty

7.4.1 SHI Input/Output Shift Register (IOSR)—Host Side

The variable length Input/Output Shift Register (IOSR) can be viewed as a serial-to-parallel and parallel-to-serial buffer in the SHI. The IOSR is involved with every data transfer in both directions (read and write). In compliance with the I^2C and SPI bus protocols, data is shifted in and out MSB first. In 8-bit data transfer modes, the most significant byte of the IOSR is used as the shift register. In 16-bit data transfer modes, the two most significant bytes become the shift register. In 24-bit transfer modes, the shift register uses all three bytes of the IOSR (see Figure 7-5).

NOTE

The IOSR cannot be accessed directly either by the host processor or by the DSP. It is fully controlled by the SHI controller logic.

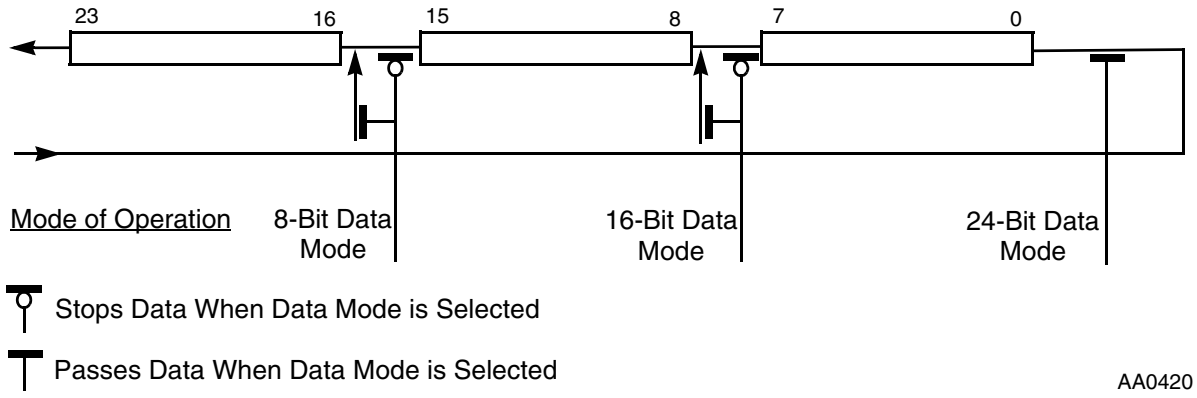


Figure 7-5 SHI I/O Shift Register (IOSR)

7.4.2 SHI Host Transmit Data Register (HTX)—DSP Side

The host transmit data register (HTX) is used for DSP-to-Host data transfers. The HTX register is 24 bits wide. Writing to the HTX register by DSP core instructions or by DMA transfers clears the HTDE flag. The DSP may program the HTIE bit to cause a host transmit data interrupt when HTDE is set (see Section 7.4.6.10, "HCSR Transmit-Interrupt Enable (HTIE)—Bit 11"). Data should not be written to the HTX until HTDE is set in order to prevent overwriting the previous data. HTX is reset to the empty state when in stop mode and during hardware reset, software reset, and individual reset.

In the 8-bit data transfer mode the most significant byte of the HTX is transmitted; in the 16-bit mode the two most significant bytes, and in the 24-bit mode all the contents of HTX is transferred.

7.4.3 SHI Host Receive Data FIFO (HRX)—DSP Side

The 24-bit host receive data FIFO (HRX) is a 10-word deep, First-In-First-Out (FIFO) register used for Host-to-DSP data transfers. The serial data is received via the shift register and then loaded into the HRX. In the 8-bit data transfer mode, the most significant byte of the shift register is transferred to the HRX (the other bits are cleared); in the 16-bit mode the two most significant bytes are transferred (the least significant byte is cleared), and in the 24-bit mode, all 24 bits are transferred to the HRX. The HRX may be read by the DSP while the FIFO is being loaded from the shift register. Reading all data from HRX clears the HRNE flag. The HRX may be read by DSP core instructions or by DMA transfers. The HRX FIFO is reset to the empty state when the chip is in stop mode, and during hardware reset, software reset, and individual reset.

7.4.4 SHI Slave Address Register (HSAR)—DSP Side

The 24-bit slave address register (HSAR) is used when the SHI operates in the I^2C slave mode and is ignored in the other operational modes. HSAR holds five bits of the 7-bit slave device address. The SHI also acknowledges the general call address specified by the I^2C protocol (eight zeroes comprising a 7-bit address and a R/$\overline{W}$ bit), but treats any following data bytes as regular data. That is, the SHI does not differentiate between its dedicated address and the general call address. HSAR cannot be accessed by the host processor.

7.4.4.1 HSAR Reserved Bits—Bits 19, 17–0

These bits are reserved. They read as zero and should be written with zero for future compatibility.

7.4.4.2 HSAR I^2C Slave Address (HA[6:3], HA1)—Bits 23–20,18

Part of the I^2C slave device address is stored in the read/write HA[6:3], HA1 bits of HSAR. The full 7-bit slave device address is formed by combining the HA[6:3], HA1 bits with the HA0 and HA2 pins to obtain the HA[6:0] slave device address. The full 7-bit slave device address is compared to the received address byte whenever an I^2C master device initiates an I^2C bus transfer. During hardware reset or software reset, HA[6:3] = 1011 and HA1 is cleared; this results in a default slave device address of 1011[HA2]0[HA0].

7.4.5 SHI Clock Control Register (HCKR)—DSP Side

The HCKR is a 24-bit read/write register that controls SHI clock generator operation. The HCKR bits should be modified only while the SHI is in the individual reset state (HEN = 0 in the HCSR).

For proper SHI clock setup, please consult the datasheet. The programmer should not use the combination HRS = 1 and HDM[7:0] = 00000000, since it may cause synchronization problems and improper operation (it is an illegal combination).

The HCKR bits are cleared during hardware reset or software reset, except for CPHA, which is set. The HCKR is not affected by the stop state.

The HCKR bits are described in the following paragraphs.

7.4.5.1 Clock Phase and Polarity (CPHA and CPOL)—Bits 1–0

The Clock Phase (CPHA) bit controls the relationship between the data on the master-in-slave-out (MISO) and master-out-slave-in (MOSI) pins and the clock produced or received at the SCK pin. The CPOL bit determines the clock polarity (1 = active-high, 0 = active-low).

The clock phase and polarity should be identical for both the master and slave SPI devices. CPHA and CPOL are functional only when the SHI operates in the SPI mode, and are ignored in the I^2C mode. The CPHA bit is set and the CPOL bit is cleared during hardware reset and software reset.

The programmer may select any of four combinations of serial clock (SCK) phase and polarity when operating in the SPI mode (See Figure 7-6).

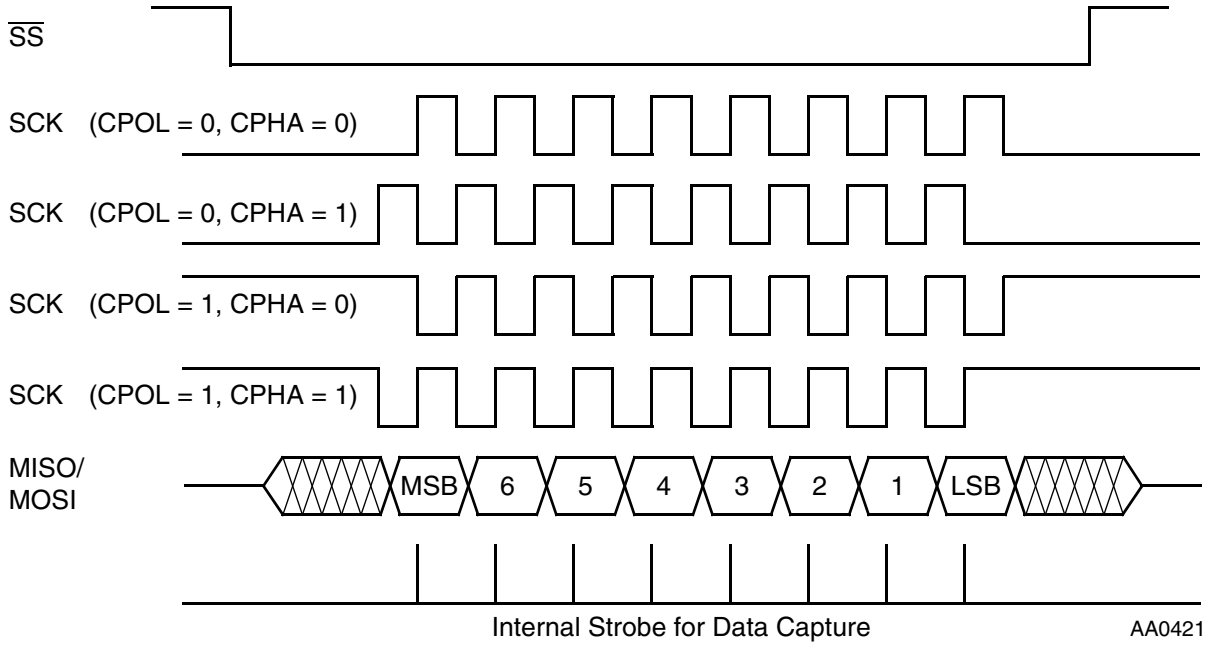


Figure 7-6 SPI Data-To-Clock Timing Diagram

If CPOL is cleared, it produces a steady-state low value at the SCK pin of the master device whenever data is not being transferred. If the CPOL bit is set, it produces a high value at the SCK pin of the master device whenever data is not being transferred.

CPHA is used with the CPOL bit to select the desired clock-to-data relationship. The CPHA bit, in general, selects the clock edge that captures data and allows it to change states. It has its greatest impact on the first bit transmitted (MSB) in that it does or does not allow a clock transition before the data capture edge.

When the SHI is in slave mode and CPHA = 0, the $\overline{SS}$ line must be deasserted and asserted by the external master between each successive word transfer. $\overline{SS}$ must remain asserted between successive bytes within a word. The DSP core should write the next data word to HTX when HTDE = 1, clearing HTDE. However, the data is transferred to the shift register for transmission only when $\overline{SS}$ is deasserted. HTDE is set when the data is transferred from HTX to the shift register.

When the SHI is in slave mode and CPHA = 1, the $\overline{SS}$ line may remain asserted between successive word transfers. The $\overline{SS}$ must remain asserted between successive bytes within a word. The DSP core should write the next data word to HTX when HTDE = 1, clearing HTDE. The HTX data is transferred to the shift register for transmission as soon as the shift register is empty. HTDE is set when the data is transferred from HTX to the shift register.

When the SHI is in master mode and CPHA = 0, the DSP core should write the next data word to HTX when HTDE = 1, clearing HTDE. The data is transferred immediately to the shift register for transmission. HTDE is set only at the end of the data word transmission.

NOTE

The master is responsible for deasserting and asserting the slave device $\overline{SS}$ line between word transmissions.

When the SHI is in master mode and CPHA = 1, the DSP core should write the next data word to HTX when HTDE = 1, clearing HTDE. The HTX data is transferred to the shift register for transmission as soon as the shift register is empty. HTDE is set when the data is transferred from HTX to the shift register.

7.4.5.2 HCKR Prescaler Rate Select (HRS)—Bit 2

The HRS bit controls a prescaler in series with the clock generator divider. This bit is used to extend the range of the divider when slower clock rates are desired. When HRS is set, the prescaler is bypassed. When HRS is cleared, the fixed divide-by-eight prescaler is operational. HRS is ignored when the SHI operates in the slave mode, except for I^2C when HCKFR is set. The HRS bit is cleared during hardware reset and software reset.

NOTE

Use the equations in the SHI datasheet to determine the value of HRS for the specific serial clock frequency required.

7.4.5.3 HCKR Divider Modulus Select (HDM[7:0])—Bits 10–3

The HDM[7:0] bits specify the divide ratio of the clock generator divider. A divide ratio between 1 and 256 (HDM[7:0] = $00 to $FF) may be selected. When the SHI operates in the slave mode, the HDM[7:0] bits are ignored (except for I^2C when HCKFR is set). The HDM[7:0] bits are cleared during hardware reset and software reset.

NOTE

Use the equations in the SHI datasheet to determine the value of HDM[7:0] for the specific serial clock frequency required.

7.4.5.4 HCKR Reserved Bits—Bits 23–14, 11

These bits in HCKR are reserved. They are read as zero and should be written with zero for future compatibility.

7.4.5.5 HCKR Filter Mode (HFM[1:0]) — Bits 13–12

The read/write control bits HFM[1:0] specify the operational mode of the noise reduction filters, as described in Table 7-3. The filters are designed to eliminate undesired spikes that might occur on the clock and data-in lines and allow the SHI to operate in noisy environments when required. One filter is located in the input path of the SCK/SCL line and the other is located in the input path of the data line (i.e., the SDA line when in I^2C mode, the MISO line when in SPI master mode, and the MOSI line when in SPI slave mode).

Table 7-3 SHI Noise Reduction Filter Mode

HFM1	HFM0	Description
0	0	Bypassed (Disabled)
0	1	Reserved
1	0	Narrow Spike Tolerance
1	1	Wide Spike Tolerance

When HFM[1:0] = 00, the filter is bypassed (spikes are **not** filtered out). This mode is useful when higher bit-rate transfers are required and the SHI operates in a noise-free environment.

When HFM[1:0] = 10, the narrow-spike-tolerance filter mode is selected. In this mode the filters eliminate spikes with durations of up to 50ns. This mode is suitable for use in mildly noisy environments and imposes some limitations on the maximum achievable bit-rate transfer.

When HFM[1:0] = 11, the wide-spike-tolerance filter mode is selected. In this mode the filters eliminate spikes up to 100 ns. This mode is recommended for use in noisy environments; the bit-rate transfer is strictly limited. The wide-spike- tolerance filter mode is highly recommended for use in I^2C bus systems as it fully conforms to the I^2C bus specification and improves noise immunity.

NOTE

HFM[1:0] are cleared during hardware reset and software reset.

After changing the filter bits in the HCKR to a non-bypass mode (HFM[1:0] not equal to ‘00’), the programmer should wait at least ten times the tolerable spike width before enabling the SHI (setting the HEN bit in the HCSR). Similarly, after changing the HI^2C bit in the HCSR or the CPOL bit in the HCKR, while the filter mode bits are in a non-bypass mode (HFM[1:0] not equal to ‘00’), the programmer should wait at least ten times the tolerable spike width before enabling the SHI (setting HEN in the HCSR).

7.4.6 SHI Control/Status Register (HCSR)—DSP Side

The HCSR is a 24-bit register that controls the SHI operation and reflects its status. The control bits are read/write. The status bits are read-only. The bits are described in the following paragraphs. When in the stop state or during individual reset, the HCSR status bits are reset to their hardware-reset state, while the control bits are not affected.

7.4.6.1 HCSR Host Enable (HEN)—Bit 0

The read/write control bit HEN, when set, enables the SHI. When HEN is cleared, the SHI is disabled (that is, it is in the individual reset state, see below). The HCKR and the HCSR control bits are not affected when HEN is cleared. When operating in master mode, HEN should be cleared only when the SHI is idle (HBUSY = 0). HEN is cleared during hardware reset and software reset.

7.4.6.1.1 SHI Individual Reset

While the SHI is in the individual reset state, SHI input pins are inhibited, output and bidirectional pins are disabled (high impedance), the HCSR status bits and the transmit/receive paths are reset to the same state produced by hardware reset or software reset. The individual reset state is entered following a one-instruction-cycle delay after clearing HEN.

7.4.6.2 HCSR I^2C/SPI Selection (HI^2C)—Bit 1

The read/write control bit HI^2C selects whether the SHI operates in the I^2C or SPI modes. When HI^2C is cleared, the SHI operates in the SPI mode. When HI^2C is set, the SHI operates in the I^2C mode. HI^2C affects the functionality of the SHI pins as described in Section 2, "Signal/Connection Descriptions". It is recommended that an SHI individual reset be generated (HEN cleared) before changing HI^2C. HI^2C is cleared during hardware reset and software reset.

7.4.6.3 HCSR Serial Host Interface Mode (HM[1:0])—Bits 3–2

The read/write control bits HM[1:0] select the size of the data words to be transferred, as shown in Table 7-4. HM[1:0] should be modified only when the SHI is idle (HBUSY = 0). HM[1:0] are cleared during hardware reset and software reset

Table 7-4 SHI Data Size

HM1	HMO	Description
0	0	8-bit data
0	1	16-bit data
1	0	24-bit data
1	1	Reserved

7.4.6.4 HCSR I^2C Clock Freeze (HCKFR)—Bit 4

The read/write control bit HCKFR determines the behavior of the SHI when the SHI is unable to service the master request, when operating in the I^2C slave mode. The HCKFR bit is used only in the I^2C slave mode; it is ignored otherwise.

If HCKFR is set, the SHI holds the clock line to GND if it is not ready to send data to the master during a read transfer or if the input FIFO is full when the master attempts to execute a write transfer. In this way, the master may detect that the slave is not ready for the requested transfer, without causing an error condition in the slave. When HCKFR is set for transmit sessions, the SHI clock generator must be programmed as if to generate the same serial clock as produced by the external master, otherwise erroneous operation may result. The programmed frequency should be in the range of 1 to 0.75 times the external clock frequency.

If HCKFR is cleared, any attempt from the master to execute a transfer when the slave is not ready results in an overrun or underrun error condition.

It is recommended that an SHI individual reset be generated (HEN cleared) before changing HCKFR. HCKFR is cleared during hardware reset and software reset.

7.4.6.5 HCSR FIFO-Enable Control (HFIFO)—Bit 5

The read/write control bit HFIFO selects the receive FIFO size. When HFIFO is cleared, the FIFO has one level. When HFIFO is set, the FIFO has 10 levels. It is recommended that an SHI individual reset be generated (HEN cleared) before changing HFIFO. HFIFO is cleared during hardware reset and software reset.

7.4.6.6 HCSR Master Mode (HMST)—Bit 6

The read/write control bit HMST determines the SHI operating mode. If HMST is set, the interface operates in the master mode. If HMST is cleared, the interface operates in the slave mode. The SHI supports a single-master configuration in both I^2C and SPI modes.

When configured as an SPI master, the SHI drives the SCK line and controls the direction of the data lines MOSI and MISO. The $\overline{SS}$ line must be held deasserted in the SPI master mode; if the $\overline{SS}$ line is asserted when the SHI is in SPI master mode, a bus error is generated (the HCSR HBER bit is set—see Section 7.4.6.18, "Host Bus Error (HBER)—Bit 21").

When configured as an I^2C master, the SHI controls the I^2C bus by generating start events, clock pulses, and stop events for transmission and reception of serial data.

It is recommended that an SHI individual reset be generated (HEN cleared) before changing HMST. HMST is cleared during hardware reset and software reset.

7.4.6.7 HCSR Host-Request Enable (HRQE[1:0])—Bits 8–7

The read/write control bits HRQE[1:0] are used to control the $\overline{HREQ}$ pin. When HRQE[1:0] are cleared, the $\overline{HREQ}$ pin is disabled and held in the high impedance state. If either of HRQE[1:0] are set and the SHI is in a master mode, the $\overline{HREQ}$ pin becomes an input controlling SCK: deasserting $\overline{HREQ}$ suspends SCK. If either of HRQE[1:0] are set and the SHI is in a slave mode, $\overline{HREQ}$ becomes an output and its operation is defined in Table 7-5. HRQE[1:0] should be changed only when the SHI is idle (HBUSY = 0). HRQE[1:0] are cleared during hardware reset and software reset.

Table 7-5 $\overline{HREQ}$ Function In SHI Slave Modes

HRQE1	HRQE0	$\overline{HREQ}$ Pin Operation
0	0	High impedance
0	1	Asserted if IOSR is ready to receive a new word
1	0	Asserted if IOSR is ready to transmit a new word
1	1	I^2C: Asserted if IOSR is ready to transmit or receive SPI: Asserted if IOSR is ready to transmit and receive

7.4.6.8 HCSR Idle (HIDLE)—Bit 9

The read/write control/status bit HIDLE is used only in the I^2C master mode; it is ignored otherwise. It is only possible to set the HIDLE bit during writes to the HCSR. HIDLE is cleared by writing to HTX. To ensure correct transmission of the slave device address byte, HIDLE should be set only when HTX is empty (HTDE = 1). After HIDLE is set, a write to HTX clears HIDLE and causes the generation of a stop event, a start event, and then the transmission of the eight MSBs of the data as the slave device address byte. While HIDLE is cleared, data written to HTX is transmitted as is. If the SHI completes transmitting a word and there is no new data in HTX, the clock is suspended after sampling ACK. If HIDLE is set when the SHI completes transmitting a word with no new data in HTX, a stop event is generated.

HIDLE determines the acknowledge that the receiver sends after correct reception of a byte. If HIDLE is cleared, the reception is acknowledged by sending a 0 bit on the SDA line at the ACK clock tick. If HIDLE is set, the reception is not acknowledged (a 1 bit is sent). It is used to signal an end-of-data to a slave transmitter by not generating an ACK on the last byte. As a result, the slave transmitter must release the SDA line to allow the master to generate the stop event. If the SHI completes receiving a word and the HRX FIFO is full, the clock is suspended before transmitting an ACK. While HIDLE is cleared the bus is busy, that is, the start event was sent but no stop event was generated. Setting HIDLE causes a stop event after receiving the current word.

HIDLE is set while the SHI is not in the I^2C master mode, while the chip is in the stop state, and during hardware reset, software reset and individual reset.

NOTE

Programmers should take care to ensure that all DMA channel service to HTX is disabled before setting HIDLE.

7.4.6.9 HCSR Bus-Error Interrupt Enable (HBIE)—Bit 10

The read/write control bit HBIE is used to enable the SHI bus-error interrupt. If HBIE is cleared, bus-error interrupts are disabled, and the HBER status bit must be polled to determine if an SHI bus error occurred. If both HBIE and HBER are set, the SHI requests an SHI bus-error interrupt service from the interrupt controller. HBIE is cleared by hardware reset and software reset.

NOTE

Clearing HBIE masks a pending bus-error interrupt only after a one instruction cycle delay. If HBIE is cleared in a long interrupt service routine, it is recommended that at least one other instruction separate the instruction that clears HBIE and the RTI instruction at the end of the interrupt service routine.

7.4.6.10 HCSR Transmit-Interrupt Enable (HTIE)—Bit 11

The read/write control bit HTIE is used to enable the SHI transmit data interrupts. If HTIE is cleared, transmit interrupts are disabled, and the HTDE status bit must be polled to determine if HTX is empty. If both HTIE and HTDE are set and HTUE is cleared, the SHI requests an SHI transmit-data interrupt service from the interrupt controller. If both HTIE and HTUE are set, the SHI requests an SHI

transmit-underrun-error interrupt service from the interrupt controller. HTIE is cleared by hardware reset and software reset.

NOTE

Clearing HTIE masks a pending transmit interrupt only after a one instruction cycle delay. If HTIE is cleared in a long interrupt service routine, it is recommended that at least one other instruction separate the instruction that clears HTIE and the RTI instruction at the end of the interrupt service routine.

7.4.6.11 HCSR Receive Interrupt Enable (HRIE[1:0])—Bits 13–12

The read/write control bits HRIE[1:0] are used to enable the SHI receive-data interrupts. If HRIE[1:0] are cleared, receive interrupts are disabled, and the HRNE and HRFF (bits 17 and 19, see below) status bits must be polled to determine if there is data in the receive FIFO. If HRIE[1:0] are not cleared, receive interrupts are generated according to Table 7-6. HRIE[1:0] are cleared by hardware and software reset.

Table 7-6 HCSR Receive Interrupt Enable Bits

HRIE[1:0]	Interrupt	Condition
00	Disabled	Not applicable
01	Receive FIFO not empty Receive Overrun Error	HRNE = 1 and HROE = 0 HROE = 1
10	Reserved	Not applicable
11	Receive FIFO full Receive Overrun Error	HRFF = 1 and HROE = 0 HROE = 1

NOTE

Clearing HRIE[1:0] masks a pending receive interrupt only after a one instruction cycle delay. If HRIE[1:0] are cleared in a long interrupt service routine, it is recommended that at least one other instruction separate the instruction that clears HRIE[1:0] and the RTI instruction at the end of the interrupt service routine.

7.4.6.12 HCSR Host Transmit Underrun Error (HTUE)—Bit 14

The read-only status bit HTUE indicates whether a transmit-underrun error occurred. Transmit-underrun errors can occur only when operating in the SPI slave mode or the I^2C slave mode when HCKFR is cleared. In a master mode, transmission takes place on demand and no underrun can occur. HTUE is set when both the shift register and the HTX register are empty and the external master begins reading the next word:

- When operating in the I^2C mode, HTUE is set in the falling edge of the ACK bit. In this case, the SHI retransmits the previously transmitted word.
- When operating in the SPI mode, HTUE is set at the first clock edge if CPHA = 1; it is set at the assertion of $\overline{SS}$ if CPHA = 0.

If a transmit interrupt occurs with HTUE set, the transmit-underrun interrupt vector is generated. If a transmit interrupt occurs with HTUE cleared, the regular transmit-data interrupt vector is generated. HTUE is cleared by reading the HCSR and then writing to the HTX register. HTUE is cleared by hardware reset, software reset, SHI individual reset, and during the stop state.

7.4.6.13 HCSR Host Transmit Data Empty (HTDE)—Bit 15

The read-only status bit HTDE indicates whether the HTX register is empty and can be written by the DSP. HTDE is set when the data word is transferred from HTX to the shift register, except in SPI master mode when CPHA = 0 (see HCKR). When in the SPI master mode with CPHA = 0, HTDE is set after the end of the data word transmission. HTDE is cleared when the DSP writes the HTX either with write instructions or DMA transfers. HTDE is set by hardware reset, software reset, SHI individual reset, and during the stop state.

7.4.6.14 HCSR Reserved Bits—Bits 23, 18 and 16

These bits are reserved. They read as zero and should be written with zero for future compatibility.

7.4.6.15 Host Receive FIFO Not Empty (HRNE)—Bit 17

The read-only status bit HRNE indicates that the Host Receive FIFO (HRX) contains at least one data word. HRNE is set when the FIFO is not empty. HRNE is cleared when HRX is read by the DSP (read instructions or DMA transfers), reducing the number of words in the FIFO to zero. HRNE is cleared during hardware reset, software reset, SHI individual reset, and during the stop state.

7.4.6.16 Host Receive FIFO Full (HRFF)—Bit 19

The read-only status bit HRFF indicates, when set, that the Host Receive FIFO (HRX) is full. HRFF is cleared when HRX is read by the DSP (read instructions or DMA transfers) and at least one place is available in the FIFO. HRFF is cleared by hardware reset, software reset, SHI individual reset, and during the stop state.

7.4.6.17 Host Receive Overrun Error (HROE)—Bit 20

The read-only status bit HROE indicates, when set, that a data-receive overrun error has occurred. Receive-overrun errors cannot occur when operating in the I^2C master mode, because the clock is suspended if the receive FIFO is full; nor can they occur in the I^2C slave mode when HCKFR is set.

HROE is set when the shift register (IOSR) is filled and ready to transfer the data word to the HRX FIFO and the FIFO is already full (HRFF is set). When a receive-overrun error occurs, the shift register is not transferred to the FIFO. If a receive interrupt occurs with HROE set, the receive-overrun interrupt vector is generated. If a receive interrupt occurs with HROE cleared, the regular receive-data interrupt vector is generated.

HROE is cleared by reading the HCSR with HROE set, followed by reading HRX. HROE is cleared by hardware reset, software reset, SHI individual reset, and during the stop state.

7.4.6.18 Host Bus Error (HBER)—Bit 21

The read-only status bit HBER indicates, when set, that an SHI bus error occurred when operating as a master (HMST set). In I^2C mode, HBER is set if the transmitter does not receive an acknowledge after a byte is transferred; then a stop event is generated and transmission is suspended. In SPI mode, HBER is set if $\overline{SS}$ is asserted; then transmission is suspended at the end of transmission of the current word. HBER is cleared only by hardware reset, software reset, SHI individual reset, and during the stop state.

7.4.6.19 HCSR Host Busy (HBUSY)—Bit 22

The read-only status bit HBUSY indicates that the I^2C bus is busy (when in the I^2C mode) or that the SHI itself is busy (when in the SPI mode). When operating in the I^2C mode, HBUSY is set after the SHI detects a start event and remains set until a stop event is detected. When operating in the slave SPI mode, HBUSY is set while $\overline{SS}$ is asserted. When operating in the master SPI mode, HBUSY is set if the HTX register is not empty or if the IOSR is not empty. HBUSY is cleared otherwise. HBUSY is cleared by hardware reset, software reset, SHI individual reset, and during the stop state.

7.5 Characteristics Of The SPI Bus

The SPI bus consists of two serial data lines (MISO and MOSI), a clock line (SCK), and a Slave Select line ($\overline{SS}$). During an SPI transfer, a byte is shifted out one data pin while a different byte is simultaneously shifted in through a second data pin. It can be viewed as two 8-bit shift registers connected together in a circular manner, with one shift register on the master side and the other on the slave side. Thus the data bytes in the master device and slave device are exchanged. The MISO and MOSI data pins are used for transmitting and receiving serial data. When the SPI is configured as a master, MISO is the master data input line, and MOSI is the master data output line. When the SPI is configured as a slave device, MISO is the slave data output line, and MOSI is the slave data input line.

Clock control logic allows a selection of clock polarity and a choice of two fundamentally different clocking protocols to accommodate most available synchronous serial peripheral devices. When the SPI is configured as a master, the control bits in the HCKR select the appropriate clock rate, as well as the desired clock polarity and phase format (see Figure 7-6).

The $\overline{SS}$ line allows selection of an individual slave SPI device; slave devices that are not selected do not interfere with SPI bus activity (i.e., they keep their MISO output pin in the high-impedance state). When the SHI is configured as an SPI master device, the $\overline{SS}$ line should be held high. If the $\overline{SS}$ line is driven low when the SHI is in SPI master mode, a bus error is generated (the HCSR HBER bit is set).

7.6 Characteristics Of The I^2C Bus

The I^2C serial bus consists of two bidirectional lines, one for data signals (SDA) and one for clock signals (SCL). Both the SDA and SCL lines must be connected to a positive supply voltage via a pull-up resistor.

NOTE

In the I^2C bus specifications, the standard mode (100 kHz clock rate) and a fast mode (400 kHz clock rate) are defined. The SHI can operate in either mode.

7.6.1 Overview

The I2C bus protocol must conform to the following rules:

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is high. Changes in the data line when the clock line is high are interpreted as control signals (see Figure 7-7).

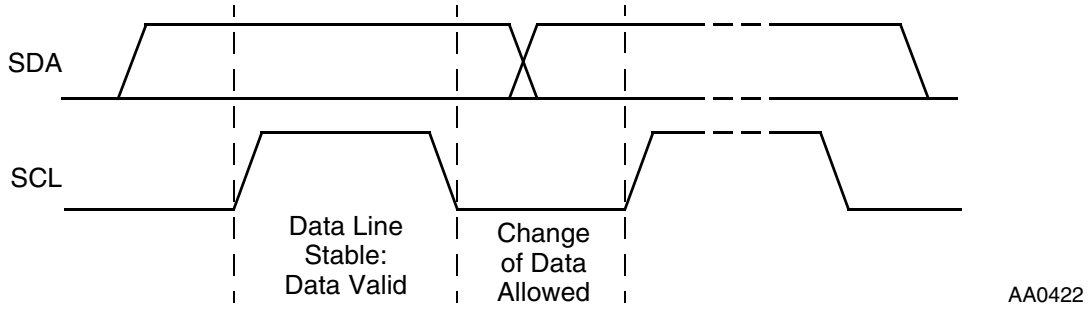


Figure 7-7 I2C Bit Transfer

Accordingly, the I2C bus protocol defines the following events:

- **Bus not busy**—Both data and clock lines remain high.
- **Start data transfer**—The start event is defined as a change in the state of the data line, from high to low, while the clock is high (see Figure 7-8).
- **Stop data transfer**—The stop event is defined as a change in the state of the data line, from low to high, while the clock is high (see Figure 7-8).
- **Data valid**—The state of the data line represents valid data when, after a start event, the data line is stable for the duration of the high period of the clock signal. The data on the line may be changed during the low period of the clock signal. There is one clock pulse per bit of data.

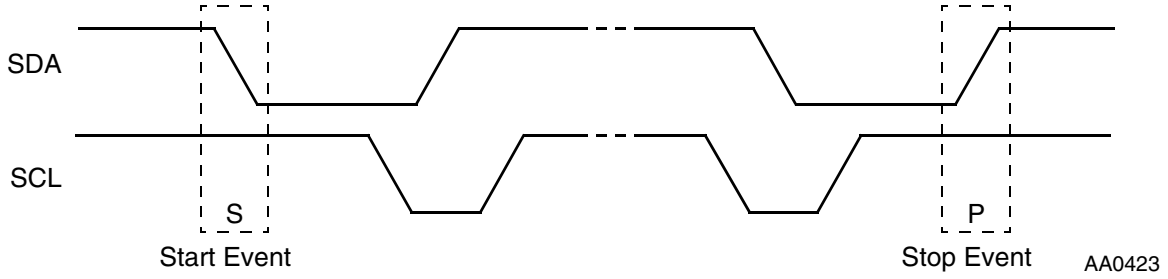


Figure 7-8 I2C Start and Stop Events

Each 8-bit word is followed by one acknowledge bit. This acknowledge bit is a high level put on the bus by the transmitter when the master device generates an extra acknowledge-related clock pulse. A slave receiver that is addressed must generate an acknowledge after each byte is received. Also, a master receiver must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The acknowledging device must pull down the SDA line during the acknowledge clock pulse so that the SDA line is stable low during the high period of the acknowledge-related clock pulse (see Figure 7-9).

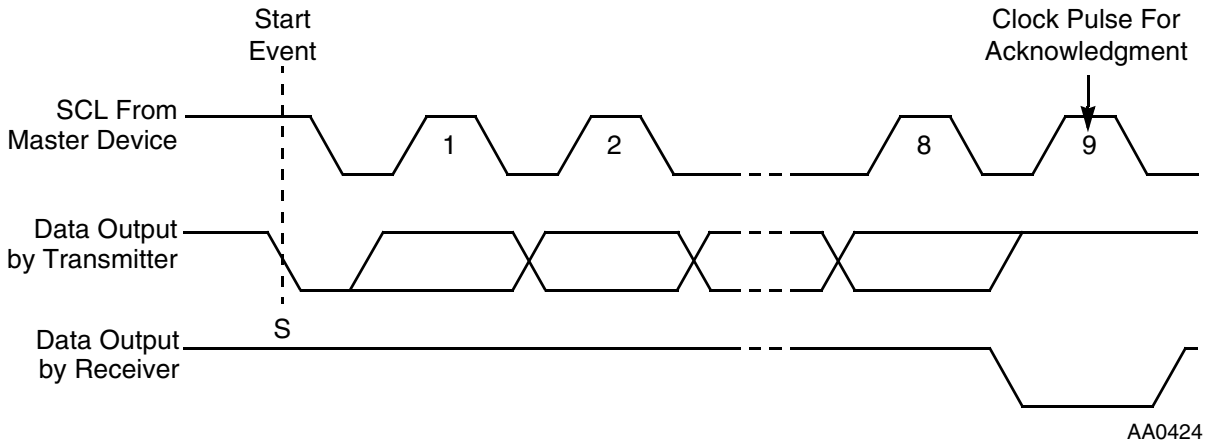


Figure 7-9 Acknowledgment on the I2C Bus

A device generating a signal is called a transmitter, and a device receiving a signal is called a receiver. A device controlling a signal is called a master and devices controlled by the master are called slaves. A master receiver must signal an end-of-data to the slave transmitter by not generating an acknowledge on the last byte clocked out of the slave device. In this case the transmitter must leave the data line high to enable the master to generate the stop event. Handshaking may also be accomplished by using the clock synchronizing mechanism. Slave devices can hold the SCL line low, after receiving and acknowledging a byte, to force the master into a wait state until the slave device is ready for the next byte transfer. The SHI supports this feature when operating as a master device and waits until the slave device releases the SCL line before proceeding with the data transfer.

7.6.2 I2C Data Transfer Formats

I2C bus data transfers follow the following process: after the start event, a slave device address is sent. The address consists of seven address bits and an eighth bit as a data direction bit (R/W). In the data direction bit, zero indicates a transmission (write), and one indicates a request for data (read). A data transfer is always terminated by a stop event generated by the master device. However, if the master device still wishes to communicate on the bus, it can generate another start event, and address another slave device without first generating a stop event (the SHI does not support this feature when operating as an I2C master device). This method is also used to provide indivisible data transfers. Various combinations of read/write formats are illustrated in Figure 7-10 and Figure 7-11.

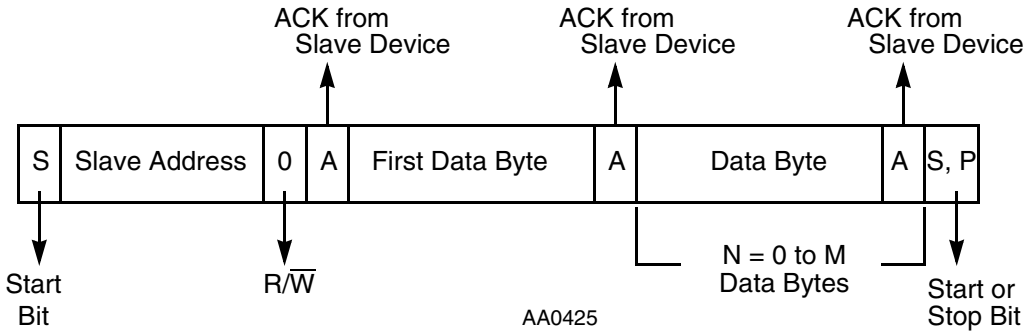


Figure 7-10 I2C Bus Protocol For Host Write Cycle

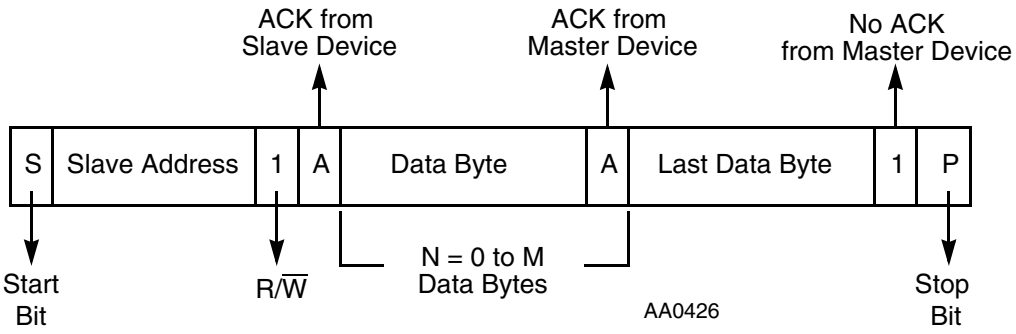


Figure 7-11 I²C Bus Protocol For Host Read Cycle

NOTE

The first data byte in a write-bus cycle can be used as a user-predefined control byte (e.g., to determine the location to which the forthcoming data bytes should be transferred).

7.7 SHI Programming Considerations

The SHI implements both SPI and I²C bus protocols and can be programmed to operate as a slave device or a single-master device. Once the operating mode is selected, the SHI may communicate with an external device by receiving and/or transmitting data. Before changing the SHI operational mode, an SHI individual reset should be generated by clearing the HEN bit. The following paragraphs describe programming considerations for each operational mode.

7.7.1 SPI Slave Mode

The SPI slave mode is entered by enabling the SHI (HEN=1), selecting the SPI mode (HI²C=0), and selecting the slave mode of operation (HMST=0). The programmer should verify that the CPHA and CPOL bits (in the HCKR) correspond to the external host clock phase and polarity. Other HCKR bits are ignored. When configured in the SPI slave mode, the SHI external pins operate as follows:

- SCK/SCL is the SCK serial clock input.
- MISO/SDA is the MISO serial data output.
- MOSI/HA0 is the MOSI serial data input.
- $\overline{SS}$/HA2 is the $\overline{SS}$ slave select input.
- $\overline{HREQ}$ is the Host Request output.

In the SPI slave mode, a receive, transmit, or full-duplex data transfer may be performed. Actually, the interface performs data receive and transmit simultaneously. The status bits of both receive and transmit paths are active; however, the programmer may disable undesired interrupts and ignore irrelevant status bits. It is recommended that an SHI individual reset (HEN cleared) be generated before beginning data reception in order to reset the HRX FIFO to its initial (empty) state (e.g., when switching from transmit to receive data).

If a write to HTX occurs, its contents are transferred to IOSR between data word transfers. The IOSR data is shifted out (via MISO) and received data is shifted in (via MOSI). The DSP may write HTX with either DSP instructions or DMA transfers if the HTDE status bit is set. If no writes to HTX occur, the contents of HTX are not transferred to IOSR, so the data shifted out when receiving is the data present in the IOSR at the time. The HRX FIFO contains valid receive data, which the DSP can read with either DSP instructions or DMA transfers (if the HRNE status bit is set).

The $\overline{\text{HREQ}}$ output pin, if enabled for receive (HRQE[1:0] = 01), is asserted when the IOSR is ready for receive and the HRX FIFO is not full; this operation guarantees that the next received data word is stored in the FIFO. The $\overline{\text{HREQ}}$ output pin, if enabled for transmit (HRQE[1:0] = 10), is asserted when the IOSR is loaded from HTX with a new data word to transfer. If $\overline{\text{HREQ}}$ is enabled for both transmit and receive (HRQE[1:0] = 11), it is asserted when the receive and transmit conditions are both true. $\overline{\text{HREQ}}$ is deasserted at the first clock pulse of the next data word transfer. The $\overline{\text{HREQ}}$ line may be used to interrupt the external master device. Connecting the $\overline{\text{HREQ}}$ line between two SHI-equipped DSPs, one operating as an SPI master device and the other as an SPI slave device, enables full hardware handshaking if operating with CPHA = 1.

The $\overline{\text{SS}}$ line should be kept asserted during a data word transfer. If the $\overline{\text{SS}}$ line is deasserted before the end of the data word transfer, the transfer is aborted and the received data word is lost.

7.7.2 SPI Master Mode

The SPI master mode is initiated by enabling the SHI (HEN = 1), selecting the SPI mode (HI^2C = 0), and selecting the master mode of operation (HMST = 1). Before enabling the SHI as an SPI master device, the programmer should program the proper clock rate, phase and polarity in HCKR. When configured in the SPI master mode, the SHI external pins operate as follows:

- SCK/SCL is the SCK serial clock output.
- MISO/SDA is the MISO serial data input.
- MOSI/HA0 is the MOSI serial data output.
- $\overline{\text{SS}}$/HA2 is the $\overline{\text{SS}}$ input. It should be kept deasserted (high) for proper operation.
- $\overline{\text{HREQ}}$ is the Host Request input.

The external slave device can be selected either by using external logic or by activating a GPIO pin connected to its $\overline{\text{SS}}$ pin. However, the $\overline{\text{SS}}$ input pin of the SPI master device should be held deasserted (high) for proper operation. If the SPI master device $\overline{\text{SS}}$ pin is asserted, the host bus error status bit (HBER) is set. If the HBIE bit is also set, the SHI issues a request to the DSP interrupt controller to service the SHI bus error interrupt.

In the SPI master mode the DSP must write to HTX to receive, transmit or perform a full-duplex data transfer. Actually, the interface performs simultaneous data receive and transmit. The status bits of both receive and transmit paths are active; however, the programmer may disable undesired interrupts and ignore irrelevant status bits. In a data transfer, the HTX is transferred to IOSR, clock pulses are generated, the IOSR data is shifted out (via MOSI) and received data is shifted in (via MISO). The DSP programmer may write HTX (if the HTDE status bit is set) with either DSP instructions or DMA transfers to initiate the transfer of the next word. The HRX FIFO contains valid receive data, which the DSP can read with either DSP instructions or DMA transfers, if the HRNE status bit is set.

It is recommended that an SHI individual reset (HEN cleared) be generated before beginning data reception in order to reset the receive FIFO to its initial (empty) state (e.g., when switching from transmit to receive data).

The $\overline{\text{HREQ}}$ input pin is ignored by the SPI master device if the HRQE[1:0] bits are cleared, and considered if any of them is set. When asserted by the slave device, $\overline{\text{HREQ}}$ indicates that the external slave device is ready for the next data transfer. As a result, the SPI master sends clock pulses for the full data word transfer. $\overline{\text{HREQ}}$ is deasserted by the external slave device at the first clock pulse of the new data transfer. When deasserted, $\overline{\text{HREQ}}$ prevents the clock generation of the next data word transfer until it is asserted again. Connecting the $\overline{\text{HREQ}}$ line between two SHI-equipped DSPs, one operating as an SPI master device and the other as an SPI slave device, enables full hardware handshaking if CPHA = 1. For CPHA = 0, $\overline{\text{HREQ}}$ should be disabled by clearing HRQE[1:0].

7.7.3 I^2C Slave Mode

The I^2C slave mode is entered by enabling the SHI (HEN=1), selecting the I^2C mode (HI^2C=1), and selecting the slave mode of operation (HMST=0). In this operational mode the contents of HCKR are ignored. When configured in the I^2C slave mode, the SHI external pins operate as follows:

- SCK/SCL is the SCL serial clock input.
- MISO/SDA is the SDA open drain serial data line.
- MOSI/HA0 is the HA0 slave device address input.
- $\overline{\text{SS}}$/HA2 is the HA2 slave device address input.
- $\overline{\text{HREQ}}$ is the Host Request output.

When the SHI is enabled and configured in the I^2C slave mode, the SHI controller inspects the SDA and SCL lines to detect a start event. Upon detection of the start event, the SHI receives the slave device address byte and enables the slave device address recognition unit. If the slave device address byte was not identified as its personal address, the SHI controller fails to acknowledge this byte by not driving low the SDA line at the ninth clock pulse (ACK = 1). However, it continues to poll the SDA and SCL lines to detect a new start event. If the personal slave device address was correctly identified, the slave device address byte is acknowledged (ACK = 0 is sent) and a receive/transmit session is initiated according to the eighth bit of the received slave device address byte (i.e., the R/$\overline{\text{W}}$ bit).

7.7.3.1 Receive Data in I^2C Slave Mode

A receive session is initiated when the personal slave device address has been correctly identified and the R/$\overline{\text{W}}$ bit of the received slave device address byte has been cleared. Following a receive initiation, data in the SDA line is shifted into IOSR MSB first. Following each received byte, an acknowledge (ACK = 0) is sent at the ninth clock pulse via the SDA line. Data is acknowledged bytewise, as required by the I^2C bus protocol, and is transferred to the HRX FIFO when the complete word (according to HM[1:0]) is filled into IOSR. It is the responsibility of the programmer to select the correct number of bytes in an I^2C frame so that they fit in a complete number of words. For this purpose, the slave device address byte does not count as part of the data; therefore, it is treated separately.

In a receive session, only the receive path is enabled and HTX to IOSR transfers are inhibited. The HRX FIFO contains valid data, which may be read by the DSP with either DSP instructions or DMA transfers (if the HRNE status bit is set).

If HCKFR is cleared, when the HRX FIFO is full and IOSR is filled, an overrun error occurs and the HROE status bit is set. In this case, the last received byte is not acknowledged (ACK=1 is sent) and the word in the IOSR is not transferred to the HRX FIFO. This may inform the external I^2C master device of the occurrence of an overrun error on the slave side. Consequently the I^2C master device may terminate this session by generating a stop event.

If HCKFR is set, when the HRX FIFO is full the SHI holds the clock line to GND not letting the master device write to IOSR, which eliminates the possibility of reaching the overrun condition.

The $\overline{\text{HREQ}}$ output pin, if enabled for receive (HRQE[1:0] = 01), is asserted when the IOSR is ready to receive and the HRX FIFO is not full; this operation guarantees that the next received data word is stored in the FIFO. $\overline{\text{HREQ}}$ is deasserted at the first clock pulse of the next received word. The $\overline{\text{HREQ}}$ line may be used to interrupt the external I^2C master device. Connecting the $\overline{\text{HREQ}}$ line between two SHI-equipped DSPs, one operating as an I^2C master device and the other as an I^2C slave device, enables full hardware handshaking.

7.7.3.2 Transmit Data In I^2C Slave Mode

A transmit session is initiated when the personal slave device address has been correctly identified and the R/$\overline{\text{W}}$ bit of the received slave device address byte has been set. Following a transmit initiation, the IOSR is loaded from HTX (assuming the latter was not empty) and its contents are shifted out, MSB first, on the SDA line. Following each transmitted byte, the SHI controller samples the SDA line at the ninth clock pulse, and inspects the ACK status. If the transmitted byte was acknowledged (ACK = 0), the SHI controller continues and transmits the next byte. However, if it was not acknowledged (ACK = 1), the transmit session is stopped and the SDA line is released. Consequently, the external master device may generate a stop event in order to terminate the session.

HTX contents are transferred to IOSR when the complete word (according to HM[1:0]) has been shifted out. It is, therefore, the responsibility of the programmer to select the correct number of bytes in an I^2C frame so that they fit in a complete number of words. For this purpose, the slave device address byte does not count as part of the data; therefore, it is treated separately.

In a transmit session, only the transmit path is enabled and the IOSR-to-HRX FIFO transfers are inhibited. When the HTX transfers its valid data word to IOSR, the HTDE status bit is set and the DSP may write a new data word to HTX with either DSP instructions or DMA transfers.

If HCKFR is cleared and both IOSR and HTX are empty when the master device attempts a transmit session, an underrun condition occurs, setting the HTUE status bit, and the previous word is retransmitted.

If HCKFR is set and both IOSR and HTX are empty when the master device attempts a transmit session, the SHI holds the clock line to GND to avoid an underrun condition.

The $\overline{\text{HREQ}}$ output pin, if enabled for transmit (HRQE[1:0] = 10), is asserted when HTX is transferred to IOSR for transmission. When asserted, $\overline{\text{HREQ}}$ indicates that the slave device is ready to transmit the next data word. $\overline{\text{HREQ}}$ is deasserted at the first clock pulse of the next transmitted data word. The $\overline{\text{HREQ}}$ line

may be used to interrupt the external I^2C master device. Connecting the $\overline{\text{HREQ}}$ line between two SHI-equipped DSPs, one operating as an I^2C master device and the other as an I^2C slave device, enables full hardware handshaking.

7.7.4 I^2C Master Mode

The I^2C master mode is entered by enabling the SHI (HEN=1), selecting the I^2C mode (HI^2C=1) and selecting the master mode of operation (HMST=1). Before enabling the SHI as an I^2C master, the programmer should program the appropriate clock rate in HCKR.

When configured in the I^2C master mode, the SHI external pins operate as follows:

- SCK/SCL is the SCL open drain serial clock output.
- MISO/SDA is the SDA open drain serial data line.
- MOSI/HA0 is the HA0 slave device address input.
- $\overline{\text{SS}}$/HA2 is the HA2 slave device address input.
- $\overline{\text{HREQ}}$ is the Host Request input.

In the I^2C master mode, a data transfer session is always initiated by the DSP by writing to the HTX register when HIDLE is set. This condition ensures that the data byte written to HTX is interpreted as being a slave address byte. This data byte must specify the slave device address to be selected and the requested data transfer direction.

NOTE

The slave address byte should be located in the high portion of the data word, whereas the middle and low portions are ignored. Only one byte (the slave address byte) is shifted out, independent of the word length defined by the HM[1:0] bits.

In order for the DSP to initiate a data transfer the following actions are to be performed:

- The DSP tests the HIDLE status bit.
- If the HIDLE status bit is set, the DSP writes the slave device address and the R/$\overline{\text{W}}$ bit to the most significant byte of HTX.
- The SHI generates a start event.
- The SHI transmits one byte only, internally samples the R/$\overline{\text{W}}$ direction bit (last bit), and accordingly initiates a receive or transmit session.
- The SHI inspects the SDA level at the ninth clock pulse to determine the ACK value. If acknowledged (ACK = 0), it starts its receive or transmit session according to the sampled R/$\overline{\text{W}}$ value. If not acknowledged (ACK = 1), the HBER status bit in HCSR is set, which causes an SHI Bus Error interrupt request if HBIE is set, and a stop event is generated.

The $\overline{\text{HREQ}}$ input pin is ignored by the I^2C master device if HRQE[1:0] are cleared, and considered if either of them is set. When asserted, $\overline{\text{HREQ}}$ indicates that the external slave device is ready for the next data transfer. As a result, the I^2C master device sends clock pulses for the full data word transfer. $\overline{\text{HREQ}}$ is deasserted by the external slave device at the first clock pulse of the next data transfer. When deasserted, $\overline{\text{HREQ}}$ prevents the clock generation of the next data word transfer until it is asserted again. Connecting

the $\overline{\text{HREQ}}$ line between two SHI-equipped DSPs, one operating as an I^2C master device and the other as an I^2C slave device, enables full hardware handshaking.

7.7.4.1 Receive Data in I^2C Master Mode

A receive session is initiated if the R/$\overline{\text{W}}$ direction bit of the transmitted slave device address byte is set. Following a receive initiation, data in the SDA line is shifted into IOSR MSB first. Following each received byte, an acknowledge (ACK = 0) is sent at the ninth clock pulse via the SDA line if the HIDLE control bit is cleared. Data is acknowledged bytewise, as required by the I^2C bus protocol, and is transferred to the HRX FIFO when the complete word (according to HM[1:0]) is filled into IOSR. It is the responsibility of the programmer to select the correct number of bytes in an I^2C frame so that they fit in a complete number of words. For this purpose, the slave device address byte does not count as part of the data; therefore, it is treated separately.

If the I^2C slave transmitter is acknowledged, it should transmit the next data byte. In order to terminate the receive session, the programmer should set the HIDLE bit at the last required data word. As a result, the last byte of the next received data word is not acknowledged, the slave transmitter releases the SDA line, and the SHI generates the stop event and terminates the session.

In a receive session, only the receive path is enabled and the HTX-to-IOSR transfers are inhibited. If the HRNE status bit is set, the HRX FIFO contains valid data, which may be read by the DSP with either DSP instructions or DMA transfers. When the HRX FIFO is full, the SHI suspends the serial clock just before acknowledge. In this case, the clock is reactivated when the FIFO is read (the SHI gives an ACK = 0 and proceeds receiving).

7.7.4.2 Transmit Data In I^2C Master Mode

A transmit session is initiated if the R/$\overline{\text{W}}$ direction bit of the transmitted slave device address byte is cleared. Following a transmit initiation, the IOSR is loaded from HTX (assuming HTX is not empty) and its contents are shifted out, MSB-first, on the SDA line. Following each transmitted byte, the SHI controller samples the SDA line at the ninth clock pulse, and inspects the ACK status. If the transmitted byte was acknowledged (ACK=0), the SHI controller continues transmitting the next byte. However, if it was not acknowledged (ACK=1), the HBER status bit is set to inform the DSP side that a bus error (or overrun, or any other exception in the slave device) has occurred. Consequently, the I^2C master device generates a stop event and terminates the session.

HTX contents are transferred to the IOSR when the complete word (according to HM[1:0]) has been shifted out. It is, therefore, the responsibility of the programmer to select the right number of bytes in an I^2C frame so that they fit in a complete number of words. Remember that for this purpose, the slave device address byte does not count as part of the data.

In a transmit session, only the transmit path is enabled and the IOSR-to-HRX FIFO transfers are inhibited. When the HTX transfers its valid data word to the IOSR, the HTDE status bit is set and the DSP may write a new data word to HTX with either DSP instructions or DMA transfers. If both IOSR and HTX are empty, the SHI suspends the serial clock until new data is written into HTX (when the SHI proceeds with the transmit session) or HIDLE is set (the SHI reactivates the clock to generate the stop event and terminate the transmit session).

7.7.5 SHI Operation During DSP Stop

The SHI operation cannot continue when the DSP is in the stop state, because no DSP clocks are active. While the DSP is in the stop state, the SHI remains in the individual reset state.

While in the individual reset state the following is true:

- If the SHI was operating in the I^2C mode, the SHI signals are disabled (high impedance state).
- If the SHI was operating in the SPI mode, the SHI signals are not affected.
- The HCSR status bits and the transmit/receive paths are reset to the same state produced by hardware reset or software reset.
- The HCSR and HCKR control bits are not affected.

NOTE

It is recommended that the SHI be disabled before entering the stop state.

NOTES

8 Enhanced Serial AUDIO Interface (ESAI)

8.1 Introduction

The Enhanced Serial Audio Interface (ESAI) provides a full-duplex serial port for serial communication with a variety of serial devices including one or more industry-standard codecs, other DSPs, microprocessors, and peripherals which implement the Freescale SPI serial protocol. The ESAI consists of independent transmitter and receiver sections, each section with its own clock generator. It is a superset of the 56300 Family ESSI peripheral and of the 56000 Family SAI peripheral.

NOTE

The DSP56366 has two ESAI modules. This section describes the ESAI, and Section 9 describes the ESAI_1. The ESAI and ESAI_1 share 4 data pins. This is described in the ESAI_1 section.

The ESAI block diagram is shown in Figure 8-1. The ESAI is named synchronous because all serial transfers are synchronized to a clock. Additional synchronization signals are used to delineate the word frames. The normal mode of operation is used to transfer data at a periodic rate, one word per period. The network mode is similar in that it is also intended for periodic transfers; however, it supports up to 32 words (time slots) per period. This mode can be used to build time division multiplexed (TDM) networks. In contrast, the on-demand mode is intended for non-periodic transfers of data and to transfer data serially at high speed when the data becomes available. This mode offers a subset of the SPI protocol.

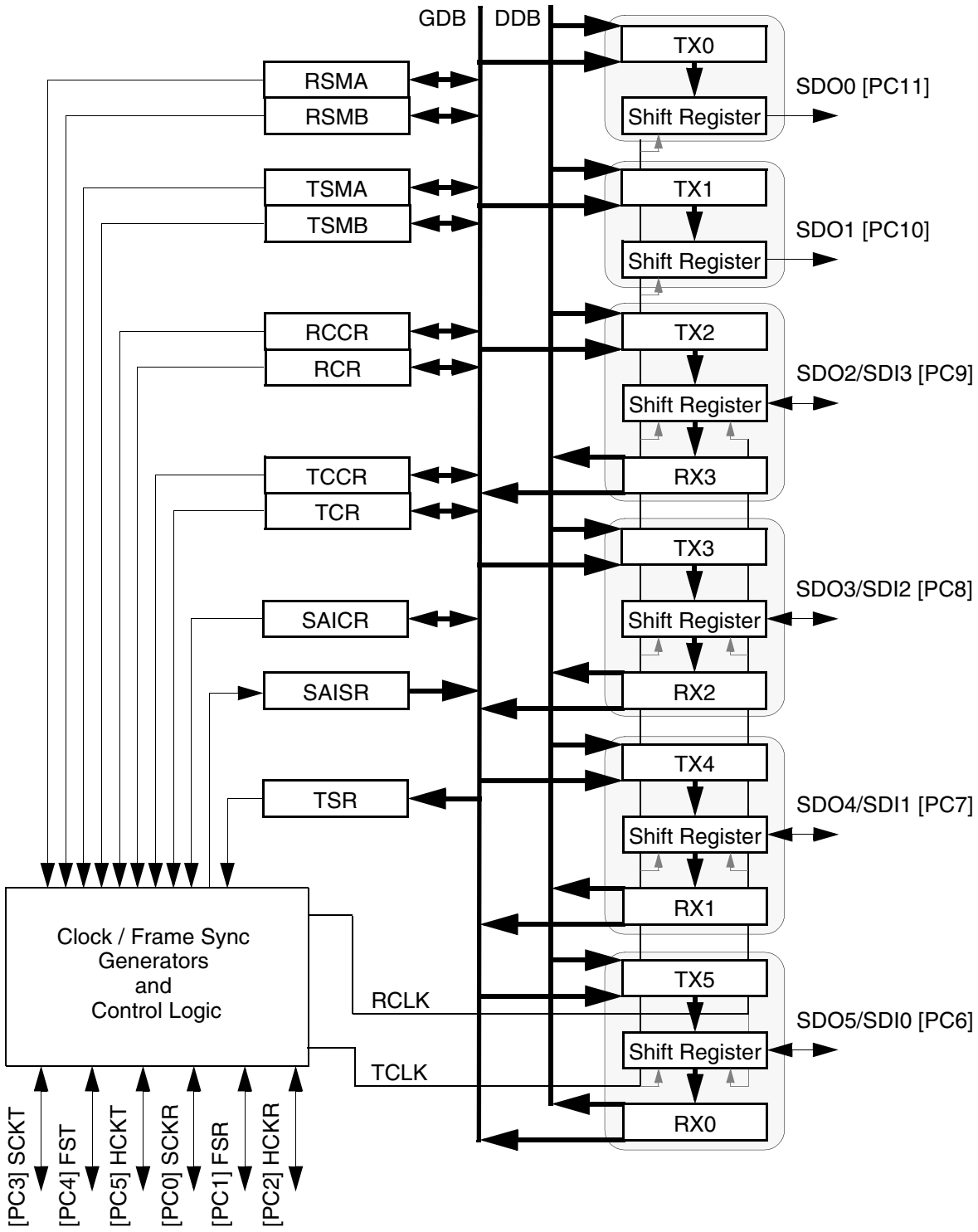


Figure 8-1 ESAI Block Diagram

8.2 ESAI Data and Control Pins

Three to twelve pins are required for operation, depending on the operating mode selected and the number of transmitters and receivers enabled. The SDO0 and SDO1 pins are used by transmitters 0 and 1 only. The SDO2/SDI3, SDO3/SDI2, SDO4/SDI1, and SDO5/SDI0 pins are shared by transmitters 2 to 5 with receivers 0 to 3. The actual mode of operation is selected under software control. All transmitters operate fully synchronized under control of the same transmitter clock signals. All receivers operate fully synchronized under control of the same receiver clock signals.

8.2.1 Serial Transmit 0 Data Pin (SDO0)

SDO0 is used for transmitting data from the TX0 serial transmit shift register. SDO0 is an output when data is being transmitted from the TX0 shift register. In the on-demand mode with an internally generated bit clock, the SDO0 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO0 may be programmed as a general-purpose I/O pin (PC11) when the ESAI SDO0 function is not being used.

8.2.2 Serial Transmit 1 Data Pin (SDO1)

SDO1 is used for transmitting data from the TX1 serial transmit shift register. SDO1 is an output when data is being transmitted from the TX1 shift register. In the on-demand mode with an internally generated bit clock, the SDO1 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO1 may be programmed as a general-purpose I/O pin (PC10) when the ESAI SDO1 function is not being used.

8.2.3 Serial Transmit 2/Receive 3 Data Pin (SDO2/SDI3)

SDO2/SDI3 is used as the SDO2 for transmitting data from the TX2 serial transmit shift register when programmed as a transmitter pin, or as the SDI3 signal for receiving serial data to the RX3 serial receive shift register when programmed as a receiver pin. SDO2/SDI3 is an input when data is being received by the RX3 shift register. SDO2/SDI3 is an output when data is being transmitted from the TX2 shift register. In the on-demand mode with an internally generated bit clock, the SDO2/SDI3 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO2/SDI3 may be programmed as a general-purpose I/O pin (PC9) when the ESAI SDO2 and SDI3 functions are not being used.

8.2.4 Serial Transmit 3/Receive 2 Data Pin (SDO3/SDI2)

SDO3/SDI2 is used as the SDO3 signal for transmitting data from the TX3 serial transmit shift register when programmed as a transmitter pin, or as the SDI2 signal for receiving serial data to the RX2 serial receive shift register when programmed as a receiver pin. SDO3/SDI2 is an input when data is being received by the RX2 shift register. SDO3/SDI2 is an output when data is being transmitted from the TX3 shift register. In the on-demand mode with an internally generated bit clock, the SDO3/SDI2 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO3/SDI2 may be programmed as a general-purpose I/O pin (PC8) when the ESAI SDO3 and SDI2 functions are not being used.

8.2.5 Serial Transmit 4/Receive 1 Data Pin (SDO4/SDI1)

SDO4/SDI1 is used as the SDO4 signal for transmitting data from the TX4 serial transmit shift register when programmed as transmitter pin, or as the SDI1 signal for receiving serial data to the RX1 serial receive shift register when programmed as a receiver pin. SDO4/SDI1 is an input when data is being received by the RX1 shift register. SDO4/SDI1 is an output when data is being transmitted from the TX4 shift register. In the on-demand mode with an internally generated bit clock, the SDO4/SDI1 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO4/SDI1 may be programmed as a general-purpose I/O pin (PC7) when the ESAI SDO4 and SDI1 functions are not being used.

8.2.6 Serial Transmit 5/Receive 0 Data Pin (SDO5/SDI0)

SDO5/SDI0 is used as the SDO5 signal for transmitting data from the TX5 serial transmit shift register when programmed as transmitter pin, or as the SDI0 signal for receiving serial data to the RX0 serial shift register when programmed as a receiver pin. SDO5/SDI0 is an input when data is being received by the RX0 shift register. SDO5/SDI0 is an output when data is being transmitted from the TX5 shift register. In the on-demand mode with an internally generated bit clock, the SDO5/SDI0 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO5/SDI0 may be programmed as a general-purpose I/O pin (PC6) when the ESAI SDO5 and SDI0 functions are not being used

8.2.7 Receiver Serial Clock (SCKR)

SCKR is a bidirectional pin providing the receivers serial bit clock for the ESAI interface. The direction of this pin is determined by the RCKD bit in the RCCR register.The SCKR operates as a clock input or output used by all the enabled receivers in the asynchronous mode (SYN=0), or as serial flag 0 pin in the synchronous mode (SYN=1).

When this pin is configured as serial flag pin, its direction is determined by the RCKD bit in the RCCR register. When configured as the output flag OF0, this pin reflects the value of the OF0 bit in the SAICR register, and the data in the OF0 bit shows up at the pin synchronized to the frame sync being used by the transmitter and receiver sections. When this pin is configured as the input flag IF0, the data value at the pin is stored in the IF0 bit in the SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.

SCKR may be programmed as a general-purpose I/O pin (PC0) when the ESAI SCKR function is not being used.

NOTE

Although the external ESAI serial clock can be independent of and asynchronous to the DSP system clock, the DSP clock frequency must be at least three times the external ESAI serial clock frequency and each ESAI serial clock phase must exceed the minimum of 1.5 DSP clock periods.

For more information on pin mode and definition, see Table 8-7 and on receiver clock signals see Table 8-1.

Table 8-1 Receiver Clock Sources (asynchronous mode only)

RHCKD	RFSD	RCKD	Receiver Bit Clock Source	OUTPUTS		
0	0	0	SCKR			
0	0	1	HCKR			SCKR
0	1	0	SCKR		FSR	
0	1	1	HCKR		FSR	SCKR
1	0	0	SCKR	HCKR		
1	0	1	INT	HCKR		SCKR
1	1	0	SCKR	HCKR	FSR	
1	1	1	INT	HCKR	FSR	SCKR

8.2.8 Transmitter Serial Clock (SCKT)

SCKT is a bidirectional pin providing the transmitters serial bit clock for the ESAI interface. The direction of this pin is determined by the TCKD bit in the TCCR register. The SCKT is a clock input or output used by all the enabled transmitters in the asynchronous mode (SYN=0) or by all the enabled transmitters and receivers in the synchronous mode (SYN=1) (see Table 8-2).

Table 8-2 Transmitter Clock Sources

THCKD	TFSD	TCKD	Transmitter Bit Clock Source	OUTPUTS		
0	0	0	SCKT			
0	0	1	HCKT			SCKT
0	1	0	SCKT		FST	
0	1	1	HCKT		FST	SCKT
1	0	0	SCKT	HCKT		
1	0	1	INT	HCKT		SCKT
1	1	0	SCKT	HCKT	FST	
1	1	1	INT	HCKT	FST	SCKT

SCKT may be programmed as a general-purpose I/O pin (PC3) when the ESAI SCKT function is not being used.

NOTE

Although the external ESAI serial clock can be independent of and asynchronous to the DSP system clock, the DSP clock frequency must be at least three times the external ESAI serial clock frequency and each ESAI serial clock phase must exceed the minimum of 1.5 DSP clock periods.

8.2.9 Frame Sync for Receiver (FSR)

FSR is a bidirectional pin providing the receivers frame sync signal for the ESAI interface. The direction of this pin is determined by the RFSD bit in RCR register. In the asynchronous mode (SYN=0), the FSR pin operates as the frame sync input or output used by all the enabled receivers. In the synchronous mode (SYN=1), it operates as either the serial flag 1 pin (TEBE=0), or as the transmitter external buffer enable control (TEBE=1, RFSD=1). For further information on pin mode and definition, see Table 8-8 and on receiver clock signals see Table 8-1.

When this pin is configured as serial flag pin, its direction is determined by the RFSD bit in the RCCR register. When configured as the output flag OF1, this pin reflects the value of the OF1 bit in the SAICR register, and the data in the OF1 bit shows up at the pin synchronized to the frame sync being used by the transmitter and receiver sections. When configured as the input flag IF1, the data value at the pin is stored in the IF1 bit in the SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.

FSR may be programmed as a general-purpose I/O pin (PC1) when the ESAI FSR function is not being used.

8.2.10 Frame Sync for Transmitter (FST)

FST is a bidirectional pin providing the frame sync for both the transmitters and receivers in the synchronous mode (SYN=1) and for the transmitters only in asynchronous mode (SYN=0) (see Table 8-2). The direction of this pin is determined by the TFSD bit in the TCR register. When configured as an output, this pin is the internally generated frame sync signal. When configured as an input, this pin receives an external frame sync signal for the transmitters (and the receivers in synchronous mode).

FST may be programmed as a general-purpose I/O pin (PC4) when the ESAI FST function is not being used.

8.2.11 High Frequency Clock for Transmitter (HCKT)

HCKT is a bidirectional pin providing the transmitters high frequency clock for the ESAI interface. The direction of this pin is determined by the THCKD bit in the TCCR register. In the asynchronous mode (SYN=0), the HCKT pin operates as the high frequency clock input or output used by all enabled transmitters. In the synchronous mode (SYN=1), it operates as the high frequency clock input or output used by all enabled transmitters and receivers. When programmed as input this pin is used as an alternative high frequency clock source to the ESAI transmitter rather than the DSP main clock. When programmed as output it can serve as a high frequency sample clock (to external DACs for example) or as an additional system clock. See Table 8-2.

HCKT may be programmed as a general-purpose I/O pin (PC5) when the ESAI HCKT function is not being used.

8.2.12 High Frequency Clock for Receiver (HCKR)

HCKR is a bidirectional pin providing the receivers high frequency clock for the ESAI interface. The direction of this pin is determined by the RHCKD bit in the RCCR register. In the asynchronous mode (SYN=0), the HCKR pin operates as the high frequency clock input or output used by all the enabled receivers. In the synchronous mode (SYN=1), it operates as the serial flag 2 pin. For further information on pin mode and definition, see Table 8-9 and on receiver clock signals see Table 8-1.

When this pin is configured as serial flag pin, its direction is determined by the RHCKD bit in the RCCR register. When configured as the output flag OF2, this pin reflects the value of the OF2 bit in the SAICR register, and the data in the OF2 bit shows up at the pin synchronized to the frame sync being used by the transmitter and receiver sections. When configured as the input flag IF2, the data value at the pin is stored in the IF2 bit in the SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.

HCKR may be programmed as a general-purpose I/O pin (PC2) when the ESAI HCKR function is not being used.

8.3 ESAI Programming Model

The ESAI can be viewed as five control registers, one status register, six transmit data registers, four receive data registers, two transmit slot mask registers, two receive slot mask registers and a

special-purpose time slot register. The following paragraphs give detailed descriptions and operations of each bit in the ESAI registers.

The ESAI pins can also function as GPIO pins (Port C), described in Section 8.5, "GPIO - Pins and Registers".

8.3.1 ESAI Transmitter Clock Control Register (TCCR)

The read/write Transmitter Clock Control Register (TCCR) controls the ESAI transmitter clock generator bit and frame sync rates, the bit clock and high frequency clock sources and the directions of the HCKT, FST and SCKT signals. (See Figure 8-2). In the synchronous mode (SYN=1), the bit clock defined for the transmitter determines the receiver bit clock as well. TCCR also controls the number of words per frame for the serial data.

	11	10	9	8	7	6	5	4	3	2	1	0
X:$FFFFB6	TDC2	TDC1	TDC0	TPSR	TPM7	TPM6	TPM5	TPM4	TPM3	TPM2	TPM1	TPM0
	23	22	21	20	19	18	17	16	15	14	13	12
	THCKD	TFSD	TCKD	THCKP	TFSP	TCKP	TFP3	TFP2	TFP1	TFP0	TDC4	TDC3

Figure 8-2 TCCR Register

Hardware and software reset clear all the bits of the TCCR register.

The TCCR control bits are described in the following paragraphs.

8.3.1.1 TCCR Transmit Prescale Modulus Select (TPM7–TPM0) - Bits 0–7

The TPM7–TPM0 bits specify the divide ratio of the prescale divider in the ESAI transmitter clock generator. A divide ratio from 1 to 256 (TPM[7:0]=$00 to $FF) may be selected. The bit clock output is available at the transmit serial bit clock (SCKT) pin of the DSP. The bit clock output is also available internally for use as the bit clock to shift the transmit and receive shift registers. The ESAI transmit clock generator functional diagram is shown in Figure 8-3.

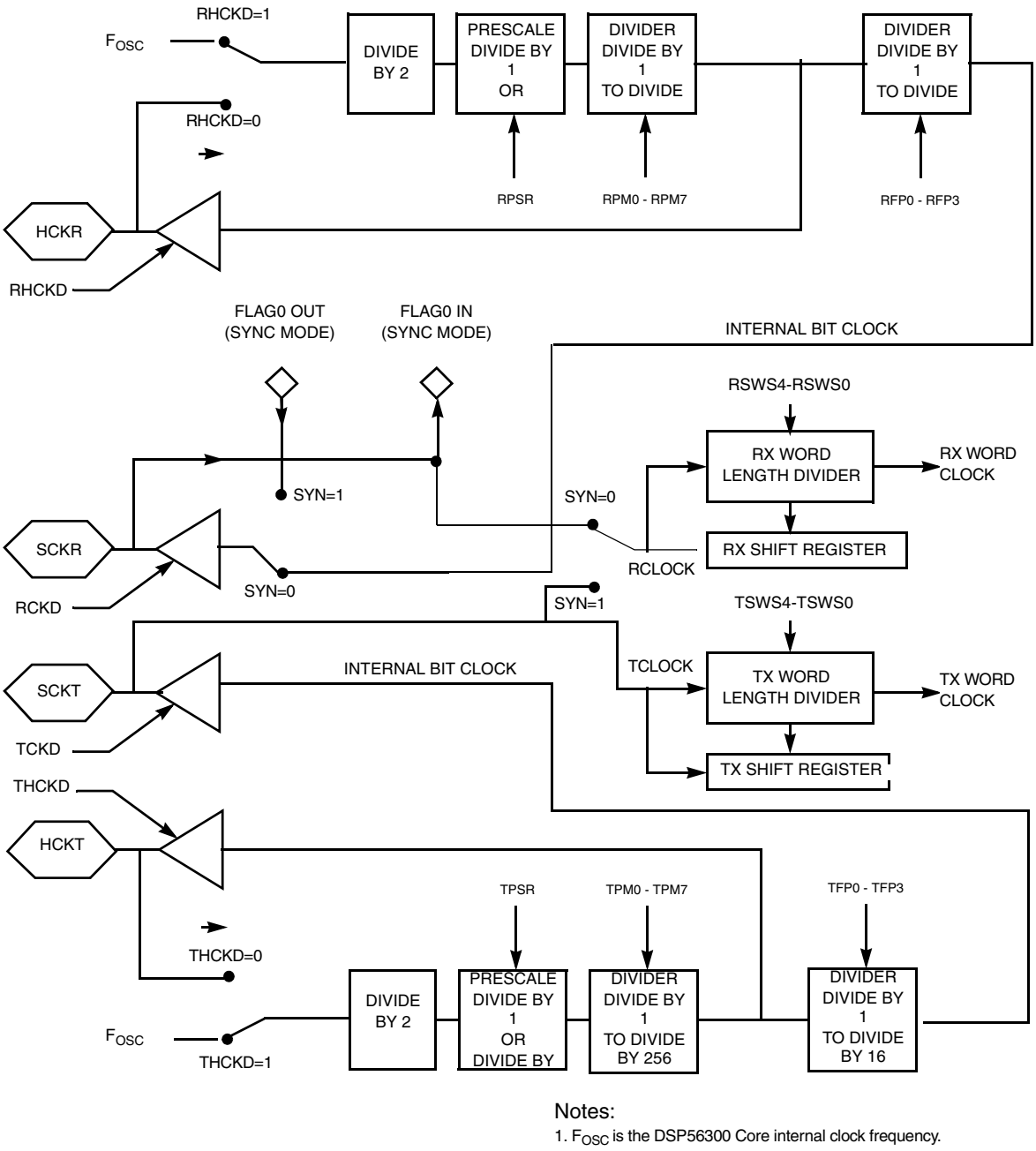


Figure 8-3 ESAI Clock Generator Functional Block Diagram

8.3.1.2 TCCR Transmit Prescaler Range (TPSR) - Bit 8

The TPSR bit controls a fixed divide-by-eight prescaler in series with the variable prescaler. This bit is used to extend the range of the prescaler for those cases where a slower bit clock is desired. When TPSR is set, the fixed prescaler is bypassed. When TPSR is cleared, the fixed divide-by-eight prescaler is

operational (see Figure 8-3). The maximum internally generated bit clock frequency is Fosc/4; the minimum internally generated bit clock frequency is Fosc/(2 x 8 x 256)=Fosc/4096.

NOTE

Do not use the combination TPSR=1 and TPM7-TPM0=$00, which causes synchronization problems when using the internal DSP clock as source (TCKD=1 or THCKD=1).

8.3.1.3 TCCR Tx Frame Rate Divider Control (TDC4–TDC0) - Bits 9–13

The TDC4–TDC0 bits control the divide ratio for the programmable frame rate dividers used to generate the transmitter frame clocks.

In network mode, this ratio may be interpreted as the number of words per frame minus one. The divide ratio may range from 2 to 32 (TDC[4:0]=00001 to 11111) for network mode. A divide ratio of one (TDC[4:0]=00000) in network mode is a special case (on-demand mode).

In normal mode, this ratio determines the word transfer rate. The divide ratio may range from 1 to 32 (TDC[4:0]=00000 to 11111) for normal mode. In normal mode, a divide ratio of 1 (TDC[4:0]=00000) provides continuous periodic data word transfers. A bit-length frame sync (TFSL=1) must be used in this case.

The ESAI frame sync generator functional diagram is shown in Figure 8-4.

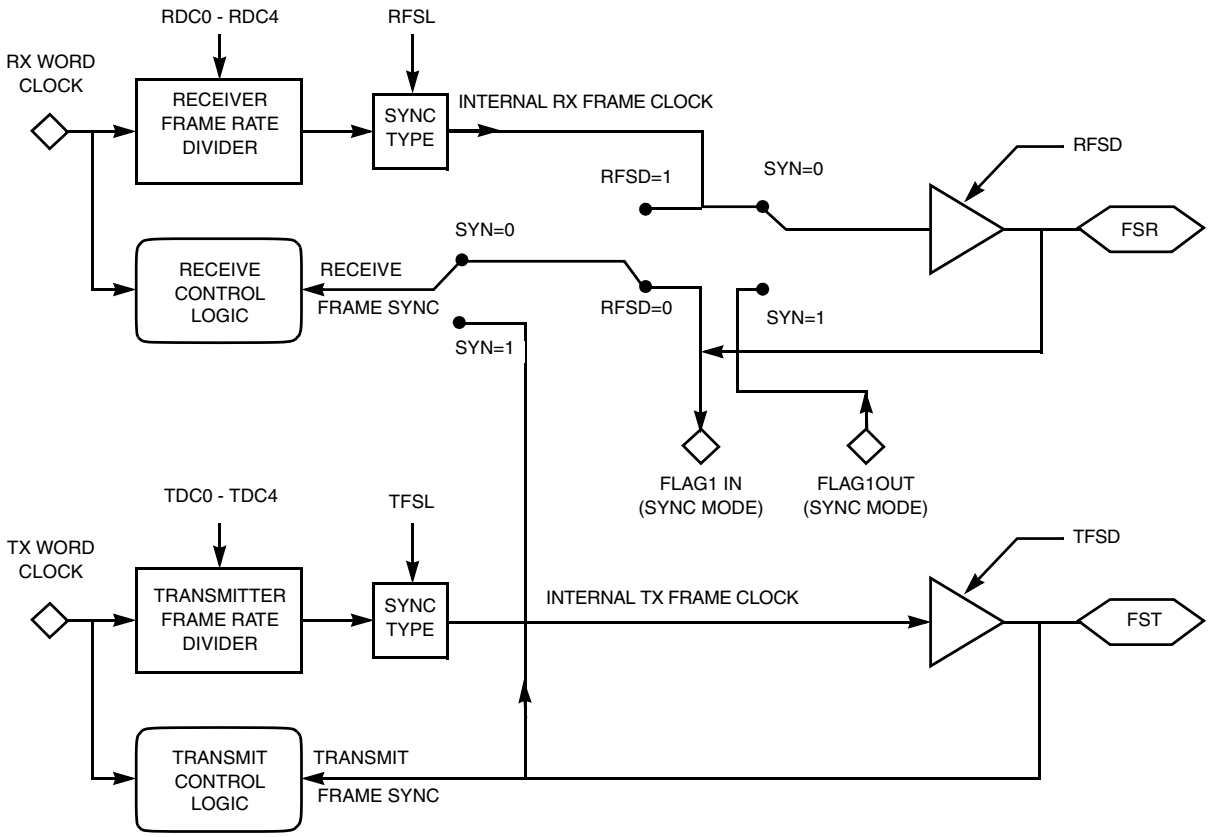


Figure 8-4 ESAI Frame Sync Generator Functional Block Diagram

8.3.1.4 TCCR Tx High Frequency Clock Divider (TFP3-TFP0) - Bits 14–17

The TFP3–TFP0 bits control the divide ratio of the transmitter high frequency clock to the transmitter serial bit clock when the source of the high frequency clock and the bit clock is the internal DSP clock. When the HCKT input is being driven from an external high frequency clock, the TFP3-TFP0 bits specify an additional division ratio in the clock divider chain. See Table 8-3 for the specification of the divide ratio. The ESAI high frequency clock generator functional diagram is shown in Figure 8-3.

Table 8-3 Transmitter High Frequency Clock Divider

TFP3-TFP0	Divide Ratio
$0	1
$1	2
$2	3
$3	4
...	...
$F	16

8.3.1.5 TCCR Transmit Clock Polarity (TCKP) - Bit 18

The Transmitter Clock Polarity (TCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If TCKP is cleared the data and the frame sync are clocked out on the rising edge of the transmit bit clock and latched in on the falling edge of the transmit bit clock. If TCKP is set the falling edge of the transmit clock is used to clock the data out and frame sync and the rising edge of the transmit clock is used to latch the data and frame sync in.

8.3.1.6 TCCR Transmit Frame Sync Polarity (TFSP) - Bit 19

The Transmitter Frame Sync Polarity (TFSP) bit determines the polarity of the transmit frame sync signal. When TFSP is cleared, the frame sync signal polarity is positive (i.e the frame start is indicated by a high level on the frame sync pin). When TFSP is set, the frame sync signal polarity is negative (i.e the frame start is indicated by a low level on the frame sync pin).

8.3.1.7 TCCR Transmit High Frequency Clock Polarity (THCKP) - Bit 20

The Transmitter High Frequency Clock Polarity (THCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If THCKP is cleared the data and the frame sync are clocked out on the rising edge of the transmit bit clock and latched in on the falling edge of the transmit bit clock. If THCKP is set the falling edge of the transmit clock is used to clock the data out and frame sync and the rising edge of the transmit clock is used to latch the data and frame sync in.

8.3.1.8 TCCR Transmit Clock Source Direction (TCKD) - Bit 21

The Transmitter Clock Source Direction (TCKD) bit selects the source of the clock signal used to clock the transmit shift registers in the asynchronous mode (SYN=0) and the transmit shift registers and the receive shift registers in the synchronous mode (SYN=1). When TCKD is set, the internal clock source becomes the bit clock for the transmit shift registers and word length divider and is the output on the SCKT pin. When TCKD is cleared, the clock source is external; the internal clock generator is disconnected from the SCKT pin, and an external clock source may drive this pin. See Table 8-2.

8.3.1.9 TCCR Transmit Frame Sync Signal Direction (TFSD) - Bit 22

TFSD controls the direction of the FST pin. When TFSD is cleared, FST is an input; when TFSD is set, FST is an output. See Table 8-2.

8.3.1.10 TCCR Transmit High Frequency Clock Direction (THCKD) - Bit 23

THCKD controls the direction of the HCKT pin. When THCKD is cleared, HCKT is an input; when THCKD is set, HCKT is an output. See Table 8-2.

8.3.2 ESAI Transmit Control Register (TCR)

The read/write Transmit Control Register (TCR) controls the ESAI transmitter section. Interrupt enable bits for the transmitter section are provided in this control register. Operating modes are also selected in this register See Figure 8-5.

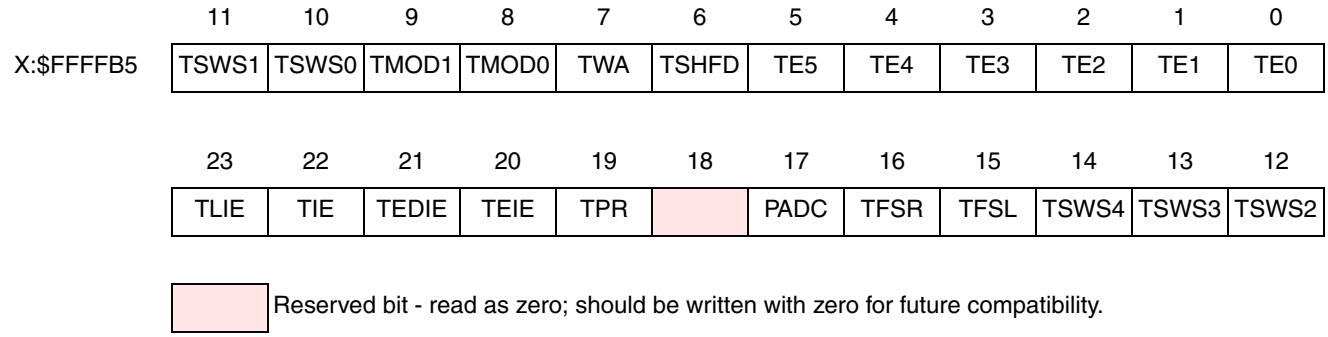

11 10 9 8 7 6 5 4 3 2 1 0
X:$FFFFB5
TSWS1 TSWS0 TMOD1 TMOD0 TWA TSHFD TE5 TE4 TE3 TE2 TE1 TE0
23 22 21 20 19 18 17 16 15 14 13 12
TLIE TIE TEDIE TEIE TPR PADC TFSR TFSL TSWS4 TSWS3 TSWS2
Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 8-5 TCR Register

Hardware and software reset clear all the bits in the TCR register.

The TCR bits are described in the following paragraphs.

8.3.2.1 TCR ESAI Transmit 0 Enable (TE0) - Bit 0

TE0 enables the transfer of data from TX0 to the transmit shift register #0. When TE0 is set and a frame sync is detected, the transmit #0 portion of the ESAI is enabled for that frame. When TE0 is cleared, the transmitter #0 is disabled after completing transmission of data currently in the ESAI transmit shift register. The SDO0 output is tri-stated, and any data present in TX0 is not transmitted (i.e., data can be written to TX0 with TE0 cleared; but data is not transferred to the transmit shift register #0).

The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE0 and setting it again disables the transmitter #0 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO0 pin remains in the high-impedance state.The on-demand mode transmit enable sequence can be the same as the normal mode, or TE0 can be left enabled.

8.3.2.2 TCR ESAI Transmit 1 Enable (TE1) - Bit 1

TE1 enables the transfer of data from TX1 to the transmit shift register #1. When TE1 is set and a frame sync is detected, the transmit #1 portion of the ESAI is enabled for that frame. When TE1 is cleared, the transmitter #1 is disabled after completing transmission of data currently in the ESAI transmit shift register. The SDO1 output is tri-stated, and any data present in TX1 is not transmitted (i.e., data can be written to TX1 with TE1 cleared; but data is not transferred to the transmit shift register #1).

The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE1 and setting it again disables the transmitter #1 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO1 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE1 can be left enabled.

8.3.2.3 TCR ESAI Transmit 2 Enable (TE2) - Bit 2

TE2 enables the transfer of data from TX2 to the transmit shift register #2. When TE2 is set and a frame sync is detected, the transmit #2 portion of the ESAI is enabled for that frame. When TE2 is cleared, the transmitter #2 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to TX2 when TE2 is cleared but the data is not transferred to the transmit shift register #2.

The SDO2/SDI3 pin is the data input pin for RX3 if TE2 is cleared and RE3 in the RCR register is set. If both RE3 and TE2 are cleared the transmitter and receiver are disabled, and the pin is tri-stated. Both RE3 and TE2 should not be set at the same time.

The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE2 and setting it again disables the transmitter #2 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO2/SDI3 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE2 can be left enabled.

8.3.2.4 TCR ESAI Transmit 3 Enable (TE3) - Bit 3

TE3 enables the transfer of data from TX3 to the transmit shift register #3. When TE3 is set and a frame sync is detected, the transmit #3 portion of the ESAI is enabled for that frame. When TE3 is cleared, the transmitter #3 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to TX3 when TE3 is cleared but the data is not transferred to the transmit shift register #3.

The SDO3/SDI2 pin is the data input pin for RX2 if TE3 is cleared and RE2 in the RCR register is set. If both RE2 and TE3 are cleared the transmitter and receiver are disabled, and the pin is tri-stated. Both RE2 and TE3 should not be set at the same time.

The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE3 and setting it again disables the transmitter #3 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO3/SDI2 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE3 can be left enabled.

8.3.2.5 TCR ESAI Transmit 4 Enable (TE4) - Bit 4

TE4 enables the transfer of data from TX4 to the transmit shift register #4. When TE4 is set and a frame sync is detected, the transmit #4 portion of the ESAI is enabled for that frame. When TE4 is cleared, the transmitter #4 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to TX4 when TE4 is cleared but the data is not transferred to the transmit shift register #4.

The SDO4/SDI1 pin is the data input pin for RX1 if TE4 is cleared and RE1 in the RCR register is set. If both RE1 and TE4 are cleared the transmitter and receiver are disabled, and the pin is tri-stated. Both RE1 and TE4 should not be set at the same time.

The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE4 and setting it again disables the transmitter #4 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO4/SDI1 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE4 can be left enabled.

8.3.2.6 TCR ESAI Transmit 5 Enable (TE5) - Bit 5

TE5 enables the transfer of data from TX5 to the transmit shift register #5. When TE5 is set and a frame sync is detected, the transmit #5 portion of the ESAI is enabled for that frame. When TE5 is cleared, the transmitter #5 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to TX5 when TE5 is cleared but the data is not transferred to the transmit shift register #5.

The SDO5/SDI0 pin is the data input pin for RX0 if TE5 is cleared and RE0 in the RCR register is set. If both RE0 and TE5 are cleared the transmitter and receiver are disabled, and the pin is tri-stated. Both RE0 and TE5 should not be set at the same time.

The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE5 and setting it again disables the transmitter #5 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO5/SDI0 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE5 can be left enabled.

8.3.2.7 TCR Transmit Shift Direction (TSHFD) - Bit 6

The TSHFD bit causes the transmit shift registers to shift data out MSB first when TSHFD equals zero or LSB first when TSHFD equals one (see Figure 8-13 and Figure 8-14).

8.3.2.8 TCR Transmit Word Alignment Control (TWA) - Bit 7

The Transmitter Word Alignment Control (TWA) bit defines the alignment of the data word in relation to the slot. This is relevant for the cases where the word length is shorter than the slot length. If TWA is cleared, the data word is left-aligned in the slot frame during transmission. If TWA is set, the data word is right-aligned in the slot frame during transmission.

Since the data word is shorter than the slot length, the data word is extended until achieving the slot length, according to the following rule:

1. If the data word is left-aligned (TWA=0), and zero padding is disabled (PADC=0), then the last data bit is repeated after the data word has been transmitted. If zero padding is enabled (PADC=1), zeroes are transmitted after the data word has been transmitted.

2. If the data word is right-aligned (TWA=1), and zero padding is disabled (PADC=0), then the first data bit is repeated before the transmission of the data word. If zero padding is enabled (PADC=1), zeroes are transmitted before the transmission of the data word.

8.3.2.9 TCR Transmit Network Mode Control (TMOD1-TMOD0) - Bits 8-9

The TMOD1 and TMOD0 bits are used to define the network mode of ESAI transmitters according to Table 8-4. In the normal mode, the frame rate divider determines the word transfer rate – one word is transferred per frame sync during the frame sync time slot, as shown in Figure 8-6. In network mode, it is possible to transfer a word for every time slot, as shown in Figure 8-6. For more details, see Section 8.4, "Operating Modes".

In order to comply with AC-97 specifications, TSWS4-TSWS0 should be set to 00011 (20-bit slot, 20-bit word length), TFSL and TFSR should be cleared, and TDC4-TDC0 should be set to $0C (13 words in frame). If TMOD[1:0]=$11 and the above recommendations are followed, the first slot and word will be 16 bits long, and the next 12 slots and words will be 20 bits long, as required by the AC97 protocol.

Table 8-4 Transmit Network Mode Selection

TMOD1	TMOD0	TDC4-TDC0	Transmitter Network Mode
0	0	$0-$1F	Normal Mode
0	1	$0	On-Demand Mode
0	1	$1-$1F	Network Mode
1	0	X	Reserved
1	1	$0C	AC97

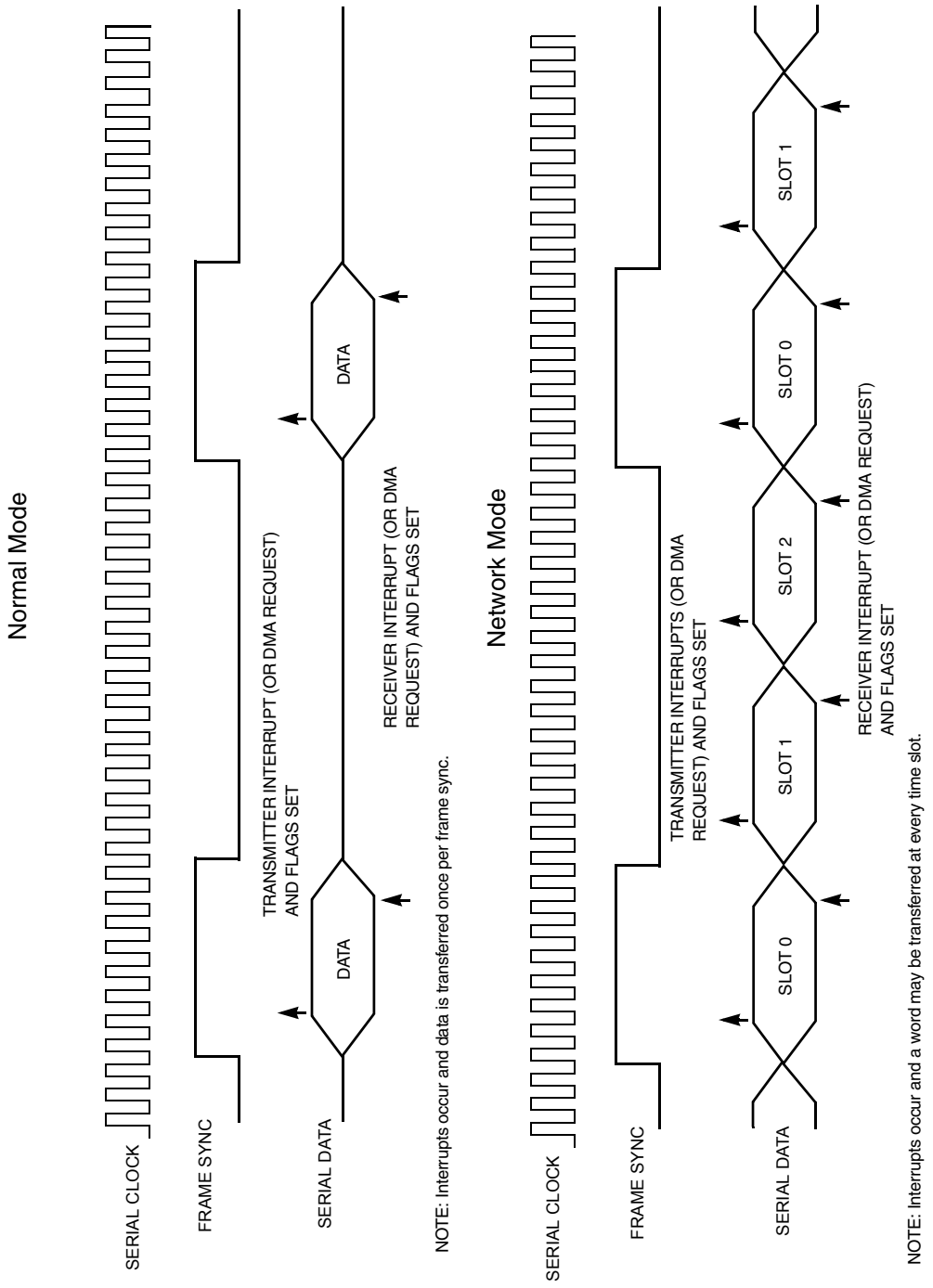

Normal Mode
SERIAL CLOCK
FRAME SYNC
SERIAL DATA
DATA
TRANSMITTER INTERRUPT (OR DMA REQUEST) AND FLAGS SET
DATA
RECEIVER INTERRUPT (OR DMA REQUEST) AND FLAGS SET
NOTE: Interrupts occur and data is transferred once per frame sync.
Network Mode
SERIAL CLOCK
FRAME SYNC
SERIAL DATA
SLOT 0
SLOT 1
TRANSMITTER INTERRUPTS (OR DMA REQUEST) AND FLAGS SET
SLOT 0
SLOT 1
SLOT 2
RECEIVER INTERRUPT (OR DMA REQUEST) AND FLAGS SET
NOTE: Interrupts occur and a word may be transferred at every time slot.

Figure 8-6 Normal and Network Operation

8.3.2.10 TCR Tx Slot and Word Length Select (TSWS4-TSWS0) - Bits 10-14

The TSWS4-TSWS0 bits are used to select the length of the slot and the length of the data words being transferred via the ESAI. The word length must be equal to or shorter than the slot length. The possible combinations are shown in Table 8-5. See also the ESAI data path programming model in Figure 8-13 and Figure 8-14.

Table 8-5 ESAI Transmit Slot and Word Length Selection

TSWS4	TSWS3	TSWS2	TSWS1	TSWS0	SLOT LENGTH	WORD LENGTH
0	0	0	0	0	8	8
0	0	1	0	0	12	8
0	0	0	0	1		12
0	1	0	0	0	16	8
0	0	1	0	1		12
0	0	0	1	0		16
0	1	1	0	0	20	8
0	1	0	0	1		12
0	0	1	1	0		16
0	0	0	1	1		20
1	0	0	0	0	24	8
0	1	1	0	1		12
0	1	0	1	0		16
0	0	1	1	1		20
1	1	1	1	0		24
1	1	0	0	0	32	8
1	0	1	0	1		12
1	0	0	1	0		16
0	1	1	1	1		20
1	1	1	1	1		24

Table 8-5 ESAI Transmit Slot and Word Length Selection (continued)

TSWS4	TSWS3	TSWS2	TSWS1	TSWS0	SLOT LENGTH	WORD LENGTH
0	1	0	1	1	Reserved	
0	1	1	1	0		
1	0	0	0	1		
1	0	0	1	1		
1	0	1	0	0		
1	0	1	1	0		
1	0	1	1	1		
1	1	0	0	1		
1	1	0	1	0		
1	1	0	1	1		
1	1	1	0	0		
1	1	1	0	1		

8.3.2.11 TCR Transmit Frame Sync Length (TFSL) - Bit 15

The TFSL bit selects the length of frame sync to be generated or recognized. If TFSL is cleared, a word-length frame sync is selected. If TFSL is set, a 1-bit clock period frame sync is selected. See Figure 8-7 for examples of frame length selection.

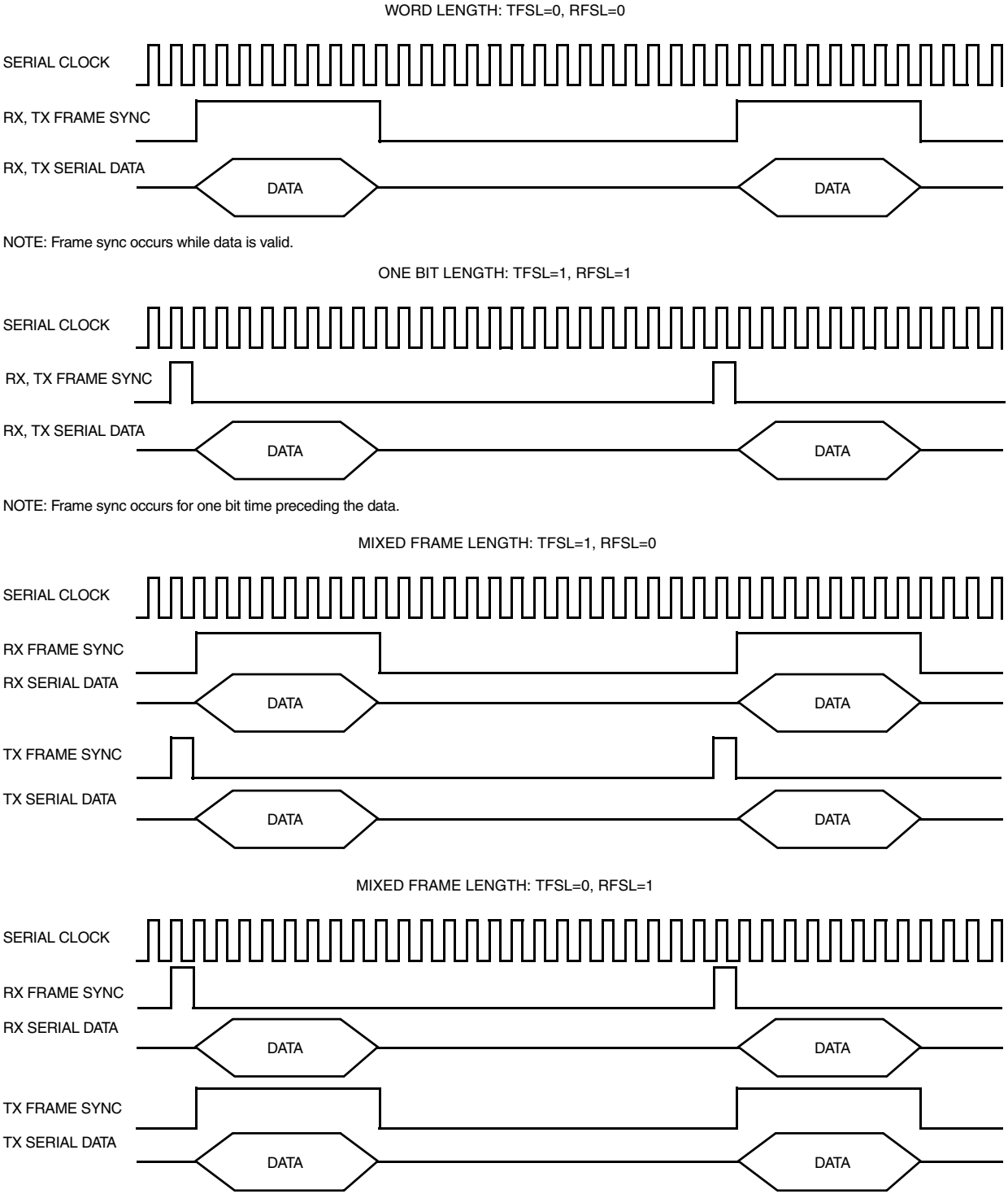


Figure 8-7 Frame Length Selection

8.3.2.12 TCR Transmit Frame Sync Relative Timing (TFSR) - Bit 16

TFSR determines the relative timing of the transmit frame sync signal as referred to the serial data lines, for a word length frame sync only (TFSL=0). When TFSR is cleared the word length frame sync occurs together with the first bit of the data word of the first slot. When TFSR is set the word length frame sync starts one serial clock cycle earlier (i.e together with the last bit of the previous data word).

8.3.2.13 TCR Transmit Zero Padding Control (PADC) - Bit 17

When PADC is cleared, zero padding is disabled. When PADC is set, zero padding is enabled. PADC, in conjunction with the TWA control bit, determines the way that padding is done for operating modes where the word length is less than the slot length. See the TWA bit description in Section 8.3.2.8, "TCR Transmit Word Alignment Control (TWA) - Bit 7" for more details.

Since the data word is shorter than the slot length, the data word is extended until achieving the slot length, according to the following rules:

1. If the data word is left-aligned (TWA=0), and zero padding is disabled (PADC=0), then the last data bit is repeated after the data word has been transmitted. If zero padding is enabled (PADC=1), zeroes are transmitted after the data word has been transmitted.
2. If the data word is right-aligned (TWA=1), and zero padding is disabled (PADC=0), then the first data bit is repeated before the transmission of the data word. If zero padding is enabled (PADC=1), zeroes are transmitted before the transmission of the data word.

8.3.2.14 TCR Reserved Bit - Bits 18

This bit is reserved. It reads as zero, and it should be written with zero for future compatibility.

8.3.2.15 TCR Transmit Section Personal Reset (TPR) - Bit 19

The TPR control bit is used to put the transmitter section of the ESAI in the personal reset state. The receiver section is not affected. When TPR is cleared, the transmitter section may operate normally. When TPR is set, the transmitter section enters the personal reset state immediately. When in the personal reset state, the status bits are reset to the same state as after hardware reset. The control bits are not affected by the personal reset state. The transmitter data pins are tri-stated while in the personal reset state; if a stable logic level is desired, the transmitter data pins should be defined as GPIO outputs, or external pull-up or pull-down resistors should be used. The transmitter clock outputs drive zeroes while in the personal reset state. Note that to leave the personal reset state by clearing TPR, the procedure described in Section 8.6, "ESAI Initialization Examples" should be followed.

8.3.2.16 TCR Transmit Exception Interrupt Enable (TEIE) - Bit 20

When TEIE is set, the DSP is interrupted when both TDE and TUE in the SAISR status register are set. When TEIE is cleared, this interrupt is disabled. Reading the SAISR status register followed by writing to all the data registers of the enabled transmitters clears TUE, thus clearing the pending interrupt.

8.3.2.17 TCR Transmit Even Slot Data Interrupt Enable (TEDIE) - Bit 21

The TEDIE control bit is used to enable the transmit even slot data interrupts. If TEDIE is set, the transmit even slot data interrupts are enabled. If TEDIE is cleared, the transmit even slot data interrupts are disabled. A transmit even slot data interrupt request is generated if TEDIE is set and the TEDE status flag in the SAISR status register is set. Even time slots are all even-numbered time slots (0, 2, 4, etc.) when operating in network mode. The zero time slot in the frame is marked by the frame sync signal and is considered to be even. Writing data to all the data registers of the enabled transmitters or to TSR clears the TEDE flag, thus servicing the interrupt.

Transmit interrupts with exception have higher priority than transmit even slot data interrupts, therefore if exception occurs (TUE is set) and TEIE is set, the ESAI requests an ESAI transmit data with exception interrupt from the interrupt controller.

8.3.2.18 TCR Transmit Interrupt Enable (TIE) - Bit 22

The DSP is interrupted when TIE and the TDE flag in the SAISR status register are set. When TIE is cleared, this interrupt is disabled. Writing data to all the data registers of the enabled transmitters or to TSR clears TDE, thus clearing the interrupt.

Transmit interrupts with exception have higher priority than normal transmit data interrupts, therefore if exception occurs (TUE is set) and TEIE is set, the ESAI requests an ESAI transmit data with exception interrupt from the interrupt controller.

8.3.2.19 TCR Transmit Last Slot Interrupt Enable (TLIE) - Bit 23

TLIE enables an interrupt at the beginning of last slot of a frame in network mode. When TLIE is set the DSP is interrupted at the start of the last slot in a frame in network mode regardless of the transmit mask register setting. When TLIE is cleared the transmit last slot interrupt is disabled. TLIE is disabled when TDC[4:0]=$00000 (on-demand mode). The use of the transmit last slot interrupt is described in Section 8.4.3, "ESAI Interrupt Requests".

8.3.3 ESAI Receive Clock Control Register (RCCR)

The read/write Receive Clock Control Register (RCCR) controls the ESAI receiver clock generator bit and frame sync rates, word length, and number of words per frame for the serial data. The RCCR control bits are described in the following paragraphs (see Figure 8-8).

	11	10	9	8	7	6	5	4	3	2	1	0
X:$FFFFB8	RDC2	RDC1	RDC0	RPSR	RPM7	RPM6	RPM5	RPM4	RPM3	RPM2	RPM1	RPM0
	23	22	21	20	19	18	17	16	15	14	13	12
	RHCKD	RFSD	RCKD	RHCKP	RFSP	RCKP	RFP3	RFP2	RFP1	RFP0	RDC4	RDC3

Figure 8-8 RCCR Register

Hardware and software reset clear all the bits of the RCCR register.

8.3.3.1 RCCR Receiver Prescale Modulus Select (RPM7–RPM0) - Bits 7–0

The RPM7–RPM0 bits specify the divide ratio of the prescale divider in the ESAI receiver clock generator. A divide ratio from 1 to 256 (RPM[7:0]=$00 to $FF) may be selected. The bit clock output is available at the receiver serial bit clock (SCKR) pin of the DSP. The bit clock output is also available internally for use as the bit clock to shift the receive shift registers. The ESAI receive clock generator functional diagram is shown in Figure 8-3.

8.3.3.2 RCCR Receiver Prescaler Range (RPSR) - Bit 8

The RPSR controls a fixed divide-by-eight prescaler in series with the variable prescaler. This bit is used to extend the range of the prescaler for those cases where a slower bit clock is desired. When RPSR is set, the fixed prescaler is bypassed. When RPSR is cleared, the fixed divide-by-eight prescaler is operational (see Figure 8-3). The maximum internally generated bit clock frequency is Fosc/4, the minimum internally generated bit clock frequency is Fosc/(2 x 8 x 256)=Fosc/4096.

NOTE

Do not use the combination RPSR=1 and RPM7-RPM0=$00, which causes synchronization problems when using the internal DSP clock as source (RHCKD=1 or RCKD=1).

8.3.3.3 RCCR Rx Frame Rate Divider Control (RDC4–RDC0) - Bits 9–13

The RDC4–RDC0 bits control the divide ratio for the programmable frame rate dividers used to generate the receiver frame clocks.

In network mode, this ratio may be interpreted as the number of words per frame minus one. The divide ratio may range from 2 to 32 (RDC[4:0]=00001 to 11111) for network mode. A divide ratio of one (RDC[4:0]=00000) in network mode is a special case (on-demand mode).

In normal mode, this ratio determines the word transfer rate. The divide ratio may range from 1 to 32 (RDC[4:0]=00000 to 11111) for normal mode. In normal mode, a divide ratio of one (RDC[4:0]=00000) provides continuous periodic data word transfers. A bit-length frame sync (RFSL=1) must be used in this case.

The ESAI frame sync generator functional diagram is shown in Figure 8-4.

8.3.3.4 RCCR Rx High Frequency Clock Divider (RFP3-RFP0) - Bits 14-17

The RFP3–RFP0 bits control the divide ratio of the receiver high frequency clock to the receiver serial bit clock when the source of the receiver high frequency clock and the bit clock is the internal DSP clock. When the HCKR input is being driven from an external high frequency clock, the RFP3-RFP0 bits specify an additional division ration in the clock divider chain. See Table 8-6 for the specification of the divide ratio. The ESAI high frequency generator functional diagram is shown in Figure 8-3.

Table 8-6 Receiver High Frequency Clock Divider

RFP3-RFP0	Divide Ratio
$0	1
$1	2
$2	3
$3	4
...	...
$F	16

8.3.3.5 RCCR Receiver Clock Polarity (RCKP) - Bit 18

The Receiver Clock Polarity (RCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If RCKP is cleared the data and the frame sync are clocked out on the rising edge of the receive bit clock and the frame sync is latched in on the falling edge of the receive bit clock. If RCKP is set the falling edge of the receive clock is used to clock the data and frame sync out and the rising edge of the receive clock is used to latch the frame sync in.

8.3.3.6 RCCR Receiver Frame Sync Polarity (RFSP) - Bit 19

The Receiver Frame Sync Polarity (RFSP) determines the polarity of the receive frame sync signal. When RFSP is cleared the frame sync signal polarity is positive (i.e the frame start is indicated by a high level on the frame sync pin). When RFSP is set the frame sync signal polarity is negative (i.e the frame start is indicated by a low level on the frame sync pin).

8.3.3.7 RCCR Receiver High Frequency Clock Polarity (RHCKP) - Bit 20

The Receiver High Frequency Clock Polarity (RHCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If RHCKP is cleared the data and the frame sync are clocked out on the rising edge of the receive bit clock and the frame sync is latched in on the falling edge of the receive bit clock. If RHCKP is set the falling edge of the receive clock is used to clock the data and frame sync out and the rising edge of the receive clock is used to latch the frame sync in.

8.3.3.8 RCCR Receiver Clock Source Direction (RCKD) - Bit 21

The Receiver Clock Source Direction (RCKD) bit selects the source of the clock signal used to clock the receive shift register in the asynchronous mode (SYN=0) and the IF0/OF0 flag direction in the synchronous mode (SYN=1).

In the asynchronous mode when RCKD is set, the internal clock source becomes the bit clock for the receive shift registers and word length divider, and is the output on the SCKR pin. In the asynchronous mode when RCKD is cleared, the clock source is external; the internal clock generator is disconnected from the SCKR pin, and an external clock source may drive this pin.

In the synchronous mode when RCKD is set, the SCKR pin becomes the OF0 output flag. If RCKD is cleared, then the SCKR pin becomes the IF0 input flag. See Table 8-1 and Table 8-7

Table 8-7 SCKR Pin Definition Table

Control Bits		SCKR PIN
SYN	RCKD	
0	0	SCKR input
0	1	SCKR output
1	0	IF0
1	1	OF0

8.3.3.9 RCCR Receiver Frame Sync Signal Direction (RFSD) - Bit 22

The Receiver Frame Sync Signal Direction (RFSD) bit selects the source of the receiver frame sync signal when in the asynchronous mode (SYN=0), and the IF1/OF1/Transmitter Buffer Enable flag direction in the synchronous mode (SYN=1).

In the asynchronous mode when RFSD is set, the internal clock generator becomes the source of the receiver frame sync, and is the output on the FSR pin. In the asynchronous mode when RFSD is cleared, the receiver frame sync source is external; the internal clock generator is disconnected from the FSR pin, and an external clock source may drive this pin.

In the synchronous mode when RFSD is set, the FSR pin becomes the OF1 output flag or the Transmitter Buffer Enable, according to the TEBE control bit. If RFSD is cleared, then the FSR pin becomes the IF1 input flag. See Table 8-1 and Table 8-8.

Table 8-8 FSR Pin Definition Table

Control Bits			FSR Pin
SYN	TEBE	RFSD	
0	X	0	FSR input
0	X	1	FSR output
1	0	0	IF1
1	0	1	OF1
1	1	0	reserved
1	1	1	Transmitter Buffer Enable

8.3.3.10 RCCR Receiver High Frequency Clock Direction (RHCKD) - Bit 23

The Receiver High Frequency Clock Direction (RHCKD) bit selects the source of the receiver high frequency clock when in the asynchronous mode (SYN=0), and the IF2/OF2 flag direction in the synchronous mode (SYN=1).

In the asynchronous mode when RHCKD is set, the internal clock generator becomes the source of the receiver high frequency clock, and is the output on the HCKR pin. In the asynchronous mode when RHCKD is cleared, the receiver high frequency clock source is external; the internal clock generator is disconnected from the HCKR pin, and an external clock source may drive this pin.

When RHCKD is cleared, HCKR is an input; when RHCKD is set, HCKR is an output.

In the synchronous mode when RHCKD is set, the HCKR pin becomes the OF2 output flag. If RHCKD is cleared, then the HCKR pin becomes the IF2 input flag. See Table 8-1 and Table 8-9.

Table 8-9 HCKR Pin Definition Table

Control Bits		HCKR PIN
SYN	RHCKD	
0	0	HCKR input
0	1	HCKR output
1	0	IF2
1	1	OF2

8.3.4 ESAI Receive Control Register (RCR)

The read/write Receive Control Register (RCR) controls the ESAI receiver section. Interrupt enable bits for the receivers are provided in this control register. The receivers are enabled in this register (0,1,2 or 3 receivers can be enabled) if the input data pin is not used by a transmitter. Operating modes are also selected in this register.

	11	10	9	8	7	6	5	4	3	2	1	0
X:$FFFFB7	RSWS1	RSWS0	RMOD	RMOD	RWA	RSHFD			RE3	RE2	RE1	RE0

23	22	21	20	19	18	17	16	15	14	13	12
RLIE	RIE	REDIE	REIE	RPR			RFSR	RFSL	RSWS4	RSWS3	RSWS2

Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 8-9 RCR Register

Hardware and software reset clear all the bits in the RCR register.

The ESAI RCR bits are described in the following paragraphs.

8.3.4.1 RCR ESAI Receiver 0 Enable (RE0) - Bit 0

When RE0 is set and TE5 is cleared, the ESAI receiver 0 is enabled and samples data at the SDO5/SDI0 pin. TX5 and RX0 should not be enabled at the same time (RE0=1 and TE5=1). When RE0 is cleared, receiver 0 is disabled by inhibiting data transfer into RX0. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the RX0 data register.

If RE0 is set while some of the other receivers are already in operation, the first data word received in RX0 will be invalid and must be discarded.

8.3.4.2 RCR ESAI Receiver 1 Enable (RE1) - Bit 1

When RE1 is set and TE4 is cleared, the ESAI receiver 1 is enabled and samples data at the SDO4/SDI1 pin. TX4 and RX1 should not be enabled at the same time (RE1=1 and TE4=1). When RE1 is cleared, receiver 1 is disabled by inhibiting data transfer into RX1. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the RX1 data register.

If RE1 is set while some of the other receivers are already in operation, the first data word received in RX1 will be invalid and must be discarded.

8.3.4.3 RCR ESAI Receiver 2 Enable (RE2) - Bit 2

When RE2 is set and TE3 is cleared, the ESAI receiver 2 is enabled and samples data at the SDO3/SDI2 pin. TX3 and RX2 should not be enabled at the same time (RE2=1 and TE3=1). When RE2 is cleared, receiver 2 is disabled by inhibiting data transfer into RX2. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the RX2 data register.

If RE2 is set while some of the other receivers are already in operation, the first data word received in RX2 will be invalid and must be discarded.

8.3.4.4 RCR ESAI Receiver 3 Enable (RE3) - Bit 3

When RE3 is set and TE2 is cleared, the ESAI receiver 3 is enabled and samples data at the SDO2/SDI3 pin. TX2 and RX3 should not be enabled at the same time (RE3=1 and TE2=1). When RE3 is cleared, receiver 3 is disabled by inhibiting data transfer into RX3. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the RX3 data register.

If RE3 is set while some of the other receivers are already in operation, the first data word received in RX3 will be invalid and must be discarded.

8.3.4.5 RCR Reserved Bits - Bits 4-5, 17-18

These bits are reserved. They read as zero, and they should be written with zero for future compatibility.

8.3.4.6 RCR Receiver Shift Direction (RSHFD) - Bit 6

The RSHFD bit causes the receiver shift registers to shift data in MSB first when RSHFD is cleared or LSB first when RSHFD is set (see Figure 8-13 and Figure 8-14).

8.3.4.7 RCR Receiver Word Alignment Control (RWA) - Bit 7

The Receiver Word Alignment Control (RWA) bit defines the alignment of the data word in relation to the slot. This is relevant for the cases where the word length is shorter than the slot length. If RWA is cleared, the data word is assumed to be left-aligned in the slot frame. If RWA is set, the data word is assumed to be right-aligned in the slot frame.

If the data word is shorter than the slot length, the data bits which are not in the data word field are ignored.

For data word lengths of less than 24 bits, the data word is right-extended with zeroes before being stored in the receive data registers.

8.3.4.8 RCR Receiver Network Mode Control (RMOD1-RMOD0) - Bits 8-9

The RMOD1 and RMOD0 bits are used to define the network mode of the ESAI receivers according to Table 8-10. In the normal mode, the frame rate divider determines the word transfer rate – one word is transferred per frame sync during the frame sync time slot, as shown in Figure 8-6. In network mode, it is possible to transfer a word for every time slot, as shown in Figure 8-6. For more details, see Section 8.4, "Operating Modes".

In order to comply with AC-97 specifications, RSWS4-RSWS0 should be set to 00011 (20-bit slot, 20-bit word), RFSL and RFSR should be cleared, and RDC4-RDC0 should be set to $0C (13 words in frame).

Table 8-10 ESAI Receive Network Mode Selection

RMOD1	RMOD0	RDC4-RDC0	Receiver Network Mode
0	0	$0-$1F	Normal Mode
0	1	$0	On-Demand Mode
0	1	$1-$1F	Network Mode
1	0	X	Reserved
1	1	$0C	AC97

8.3.4.9 RCR Receiver Slot and Word Select (RSWS4-RSWS0) - Bits 10-14

The RSWS4-RSWS0 bits are used to select the length of the slot and the length of the data words being received via the ESAI. The word length must be equal to or shorter than the slot length. The possible combinations are shown in Table 8-11. See also the ESAI data path programming model in Figure 8-13 and Figure 8-14.

Table 8-11 ESAI Receive Slot and Word Length Selection

RSWS4	RSWS3	RSWS2	RSWS1	RSWS0	SLOT LENGTH	WORD LENGTH
0	0	0	0	0	8	8
0	0	1	0	0	12	8
0	0	0	0	1		12
0	1	0	0	0	16	8
0	0	1	0	1		12
0	0	0	1	0		16
0	1	1	0	0	20	8
0	1	0	0	1		12
0	0	1	1	0		16
0	0	0	1	1		20
1	0	0	0	0	24	8
0	1	1	0	1		12
0	1	0	1	0		16
0	0	1	1	1		20
1	1	1	1	0		24
1	1	0	0	0	32	8
1	0	1	0	1		12
1	0	0	1	0		16
0	1	1	1	1		20
1	1	1	1	1		24

Table 8-11 ESAI Receive Slot and Word Length Selection (continued)

RSWS4	RSWS3	RSWS2	RSWS1	RSWS0	SLOT LENGTH	WORD LENGTH
0	1	0	1	1	Reserved	
0	1	1	1	0		
1	0	0	0	1		
1	0	0	1	1		
1	0	1	0	0		
1	0	1	1	0		
1	0	1	1	1		
1	1	0	0	1		
1	1	0	1	0		
1	1	0	1	1		
1	1	1	0	0		
1	1	1	0	1		

8.3.4.10 RCR Receiver Frame Sync Length (RFSL) - Bit 15

The RFSL bit selects the length of the receive frame sync to be generated or recognized. If RFSL is cleared, a word-length frame sync is selected. If RFSL is set, a 1-bit clock period frame sync is selected. See Figure 8-7 for examples of frame length selection.

8.3.4.11 RCR Receiver Frame Sync Relative Timing (RFSR) - Bit 16

RFSR determines the relative timing of the receive frame sync signal as referred to the serial data lines, for a word length frame sync only. When RFSR is cleared the word length frame sync occurs together with the first bit of the data word of the first slot. When RFSR is set the word length frame sync starts one serial clock cycle earlier (i.e. together with the last bit of the previous data word).

8.3.4.12 RCR Receiver Section Personal Reset (RPR) - Bit 19

The RPR control bit is used to put the receiver section of the ESAI in the personal reset state. The transmitter section is not affected. When RPR is cleared, the receiver section may operate normally. When RPR is set, the receiver section enters the personal reset state immediately. When in the personal reset state, the status bits are reset to the same state as after hardware reset.The control bits are not affected by the personal reset state.The receiver data pins are disconnected while in the personal reset state. Note that to leave the personal reset state by clearing RPR, the procedure described in Section 8.6, "ESAI Initialization Examples" should be followed.

8.3.4.13 RCR Receive Exception Interrupt Enable (REIE) - Bit 20

When REIE is set, the DSP is interrupted when both RDF and ROE in the SAISR status register are set. When REIE is cleared, this interrupt is disabled. Reading the SAISR status register followed by reading the enabled receivers data registers clears ROE, thus clearing the pending interrupt.

8.3.4.14 RCR Receive Even Slot Data Interrupt Enable (REDIE) - Bit 21

The REDIE control bit is used to enable the receive even slot data interrupts. If REDIE is set, the receive even slot data interrupts are enabled. If REDIE is cleared, the receive even slot data interrupts are disabled. A receive even slot data interrupt request is generated if REDIE is set and the REDF status flag in the SAISR status register is set. Even time slots are all even-numbered time slots (0, 2, 4, etc.) when operating in network mode. The zero time slot is marked by the frame sync signal and is considered to be even. Reading all the data registers of the enabled receivers clears the REDF flag, thus servicing the interrupt.

Receive interrupts with exception have higher priority than receive even slot data interrupts, therefore if exception occurs (ROE is set) and REIE is set, the ESAI requests an ESAI receive data with exception interrupt from the interrupt controller.

8.3.4.15 RCR Receive Interrupt Enable (RIE) - Bit 22

The DSP is interrupted when RIE and the RDF flag in the SAISR status register are set. When RIE is cleared, this interrupt is disabled. Reading the receive data registers of the enabled receivers clears RDF, thus clearing the interrupt.

Receive interrupts with exception have higher priority than normal receive data interrupts, therefore if exception occurs (ROE is set) and REIE is set, the ESAI requests an ESAI receive data with exception interrupt from the interrupt controller.

8.3.4.16 RCR Receive Last Slot Interrupt Enable (RLIE) - Bit 23

RLIE enables an interrupt after the last slot of a frame ended in network mode only. When RLIE is set the DSP is interrupted after the last slot in a frame ended regardless of the receive mask register setting. When RLIE is cleared the receive last slot interrupt is disabled. Hardware and software reset clear RLIE. RLIE is disabled when RDC[4:0]=00000 (on-demand mode). The use of the receive last slot interrupt is described in Section 8.4.3, "ESAI Interrupt Requests".

8.3.5 ESAI Common Control Register (SAICR)

The read/write Common Control Register (SAICR) contains control bits for functions that affect both the receive and transmit sections of the ESAI.See Figure 8-10.

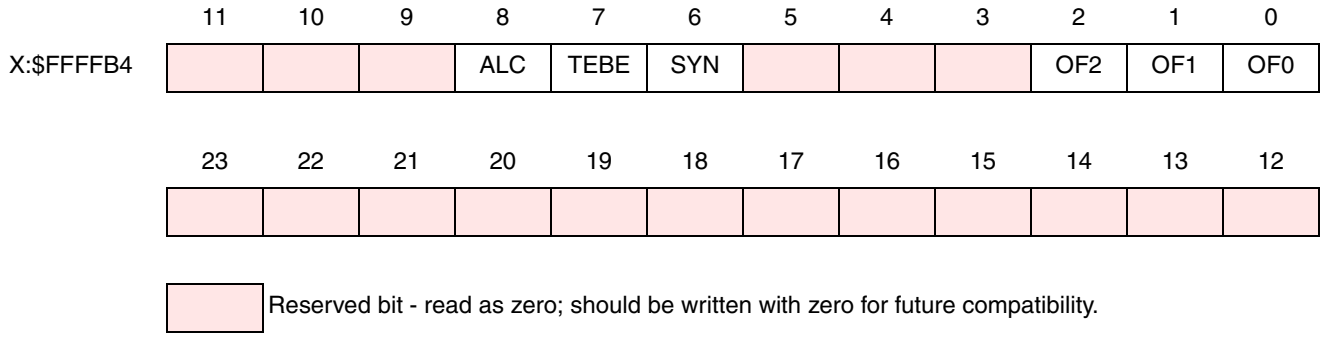

11 10 9 8 7 6 5 4 3 2 1 0
X:$FFFFB4
ALC
TEBE
SYN
OF2
OF1
OF0
23 22 21 20 19 18 17 16 15 14 13 12
Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 8-10 SAICR Register

Hardware and software reset clear all the bits in the SAICR register.

8.3.5.1 SAICR Serial Output Flag 0 (OF0) - Bit 0

The Serial Output Flag 0 (OF0) is a data bit used to hold data to be send to the OF0 pin. When the ESAI is in the synchronous clock mode (SYN=1), the SCKR pin is configured as the ESAI flag 0. If the receiver serial clock direction bit (RCKD) is set, the SCKR pin is the output flag OF0, and data present in the OF0 bit is written to the OF0 pin at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode.

8.3.5.2 SAICR Serial Output Flag 1 (OF1) - Bit 1

The Serial Output Flag 1 (OF1) is a data bit used to hold data to be send to the OF1 pin. When the ESAI is in the synchronous clock mode (SYN=1), the FSR pin is configured as the ESAI flag 1. If the receiver frame sync direction bit (RFSD) is set and the TEBE bit is cleared, the FSR pin is the output flag OF1, and data present in the OF1 bit is written to the OF1 pin at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode.

8.3.5.3 SAICR Serial Output Flag 2 (OF2) - Bit 2

The Serial Output Flag 2 (OF2) is a data bit used to hold data to be send to the OF2 pin. When the ESAI is in the synchronous clock mode (SYN=1), the HCKR pin is configured as the ESAI flag 2. If the receiver high frequency clock direction bit (RHCKD) is set, the HCKR pin is the output flag OF2, and data present in the OF2 bit is written to the OF2 pin at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode.

8.3.5.4 SAICR Reserved Bits - Bits 3-5, 9-23

These bits are reserved. They read as zero, and they should be written with zero for future compatibility.

8.3.5.5 SAICR Synchronous Mode Selection (SYN) - Bit 6

The Synchronous Mode Selection (SYN) bit controls whether the receiver and transmitter sections of the ESAI operate synchronously or asynchronously with respect to each other (see Figure 8-11). When SYN is cleared, the asynchronous mode is chosen and independent clock and frame sync signals are used for

the transmit and receive sections. When SYN is set, the synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals.

When in the synchronous mode (SYN=1), the transmit and receive sections use the transmitter section clock generator as the source of the clock and frame sync for both sections. Also, the receiver clock pins SCKR, FSR and HCKR now operate as I/O flags. See Table 8-7, Table 8-8 and Table 8-9 for the effects of SYN on the receiver clock pins.

8.3.5.6 SAICR Transmit External Buffer Enable (TEBE) - Bit 7

The Transmitter External Buffer Enable (TEBE) bit controls the function of the FSR pin when in the synchronous mode. If the ESAI is configured for operation in the synchronous mode (SYN=1), and TEBE is set while FSR pin is configured as an output (RFSD=1), the FSR pin functions as the transmitter external buffer enable control, to enable the use of an external buffers on the transmitter outputs. If TEBE is cleared then the FSR pin functions as the serial I/O flag 1. See Table 8-8 for a summary of the effects of TEBE on the FSR pin.

8.3.5.7 SAICR Alignment Control (ALC) - Bit 8

The ESAI is designed for 24-bit fractional data, thus shorter data words are left aligned to the MSB (bit 23). Some applications use 16-bit fractional data. In those cases, shorter data words may be left aligned to bit 15. The Alignment Control (ALC) bit supports these applications.

If ALC is set, transmitted and received words are left aligned to bit 15 in the transmit and receive shift registers. If ALC is cleared, transmitted and received word are left aligned to bit 23 in the transmit and receive shift registers.

NOTE

While ALC is set, 20-bit and 24-bit words may not be used, and word length control should specify 8-, 12- or 16-bit words, otherwise results are unpredictable.

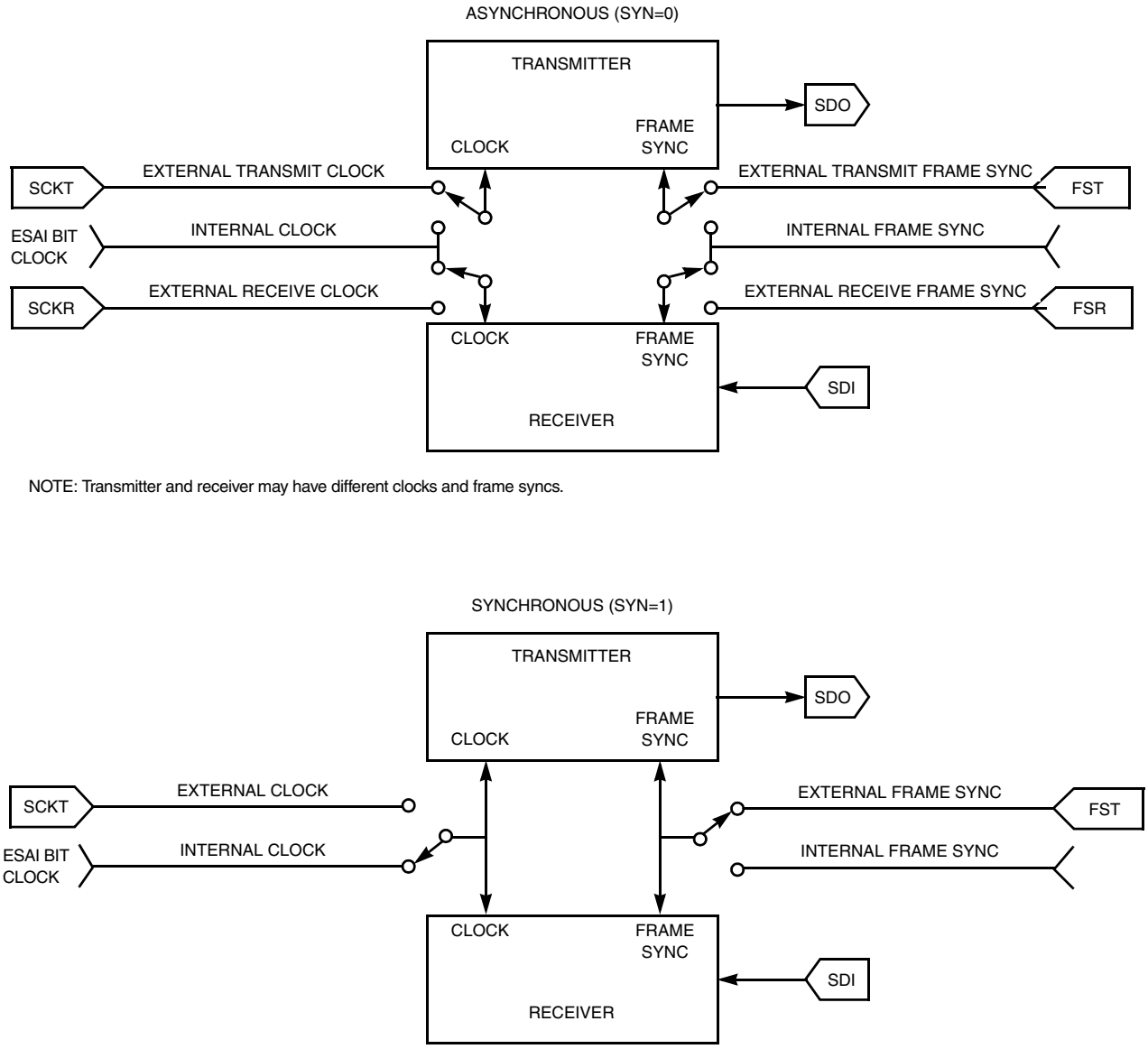


Figure 8-11 SAICR SYN Bit Operation

8.3.6 ESAI Status Register (SAISR)

The Status Register (SAISR) is a read-only status register used by the DSP to read the status and serial input flags of the ESAI. See Figure 8-12. The status bits are described in the following paragraphs.

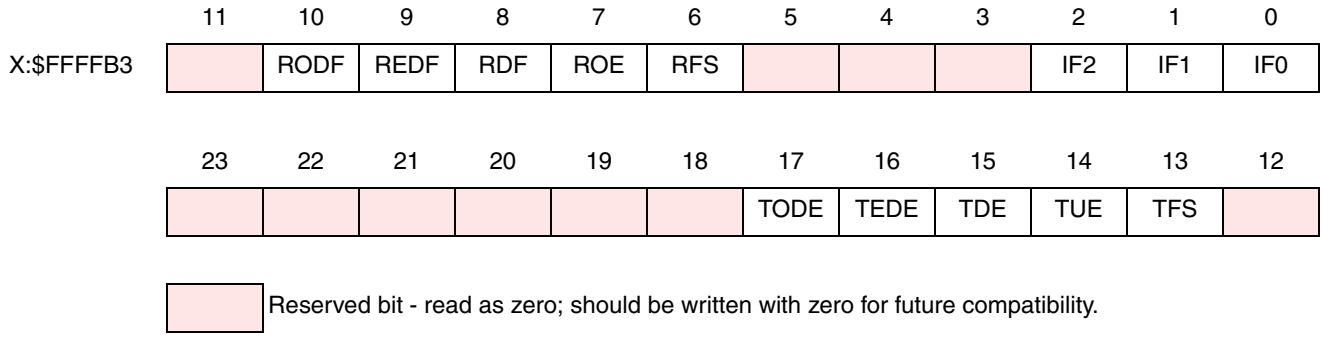

11 10 9 8 7 6 5 4 3 2 1 0
X:$FFFFB3
RODF REDF RDF ROE RFS IF2 IF1 IF0
23 22 21 20 19 18 17 16 15 14 13 12
TODE TEDE TDE TUE TFS
Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 8-12 SAISR Register

8.3.6.1 SAISR Serial Input Flag 0 (IF0) - Bit 0

The IF0 bit is enabled only when the SCKR pin is defined as ESAI in the Port Control Register, SYN=1 and RCKD=0, indicating that SCKR is an input flag and the synchronous mode is selected. Data present on the SCKR pin is latched during reception of the first received data bit after frame sync is detected. The IF0 bit is updated with this data when the receiver shift registers are transferred into the receiver data registers. IF0 reads as a zero when it is not enabled. Hardware, software, ESAI individual, and STOP reset clear IF0.

8.3.6.2 SAISR Serial Input Flag 1 (IF1) - Bit 1

The IF1 bit is enabled only when the FSR pin is defined as ESAI in the Port Control Register, SYN =1, RFSD=0 and TEBE=0, indicating that FSR is an input flag and the synchronous mode is selected. Data present on the FSR pin is latched during reception of the first received data bit after frame sync is detected. The IF1 bit is updated with this data when the receiver shift registers are transferred into the receiver data registers. IF1 reads as a zero when it is not enabled. Hardware, software, ESAI individual, and STOP reset clear IF1.

8.3.6.3 SAISR Serial Input Flag 2 (IF2) - Bit 2

The IF2 bit is enabled only when the HCKR pin is defined as ESAI in the Port Control Register, SYN=1 and RHCKD=0, indicating that HCKR is an input flag and the synchronous mode is selected. Data present on the HCKR pin is latched during reception of the first received data bit after frame sync is detected. The IF2 bit is updated with this data when the receive shift registers are transferred into the receiver data registers. IF2 reads as a zero when it is not enabled. Hardware, software, ESAI individual, and STOP reset clear IF2.

8.3.6.4 SAISR Reserved Bits - Bits 3-5, 11-12, 18-23

These bits are reserved for future use. They read as zero.

8.3.6.5 SAISR Receive Frame Sync Flag (RFS) - Bit 6

When set, RFS indicates that a receive frame sync occurred during reception of the words in the receiver data registers. This indicates that the data words are from the first slot in the frame. When RFS is clear and

a word is received, it indicates (only in the network mode) that the frame sync did not occur during reception of that word. RFS is cleared by hardware, software, ESAI individual, or STOP reset. RFS is valid only if at least one of the receivers is enabled (REx=1).

NOTE

In normal mode, RFS always reads as a one when reading data because there is only one time slot per frame – the "frame sync" time slot.

8.3.6.6 SAISR Receiver Overrun Error Flag (ROE) - Bit 7

The ROE flag is set when the serial receive shift register of an enabled receiver is full and ready to transfer to its receiver data register (RXx) and the register is already full (RDF=1). If REIE is set, an ESAI receive data with exception (overrun error) interrupt request is issued when ROE is set. Hardware, software, ESAI individual, and STOP reset clear ROE. ROE is also cleared by reading the SAISR with ROE set, followed by reading all the enabled receive data registers.

8.3.6.7 SAISR Receive Data Register Full (RDF) - Bit 8

RDF is set when the contents of the receive shift register of an enabled receiver is transferred to the respective receive data register. RDF is cleared when the DSP reads the receive data register of all enabled receivers or cleared by hardware, software, ESAI individual, or STOP reset. If RIE is set, an ESAI receive data interrupt request is issued when RDF is set.

8.3.6.8 SAISR Receive Even-Data Register Full (REDF) - Bit 9

When set, REDF indicates that the received data in the receive data registers of the enabled receivers have arrived during an even time slot when operating in the network mode. Even time slots are all even-numbered slots (0, 2, 4, 6, etc.). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. The zero time slot is considered even. REDF is set when the contents of the receive shift registers are transferred to the receive data registers. REDF is cleared when the DSP reads all the enabled receive data registers or cleared by hardware, software, ESAI individual, or STOP resets. If REDIE is set, an ESAI receive even slot data interrupt request is issued when REDF is set.

8.3.6.9 SAISR Receive Odd-Data Register Full (RODF) - Bit 10

When set, RODF indicates that the received data in the receive data registers of the enabled receivers have arrived during an odd time slot when operating in the network mode. Odd time slots are all odd-numbered slots (1, 3, 5, etc.). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. RODF is set when the contents of the receive shift registers are transferred to the receive data registers. RODF is cleared when the DSP reads all the enabled receive data registers or cleared by hardware, software, ESAI individual, or STOP resets.

8.3.6.10 SAISR Transmit Frame Sync Flag (TFS) - Bit 13

When set, TFS indicates that a transmit frame sync occurred in the current time slot. TFS is set at the start of the first time slot in the frame and cleared during all other time slots. Data written to a transmit data register during the time slot when TFS is set is transmitted (in network mode), if the transmitter is enabled,

during the second time slot in the frame. TFS is useful in network mode to identify the start of a frame. TFS is cleared by hardware, software, ESAI individual, or STOP reset. TFS is valid only if at least one transmitter is enabled (i.e. one or more of TE0, TE1, TE2, TE3, TE4 and TE5 are set).

NOTE

In normal mode, TFS always reads as a one when transmitting data because there is only one time slot per frame – the "frame sync" time slot.

8.3.6.11 SAISR Transmit Underrun Error Flag (TUE) - Bit 14

TUE is set when at least one of the enabled serial transmit shift registers is empty (no new data to be transmitted) and a transmit time slot occurs. When a transmit underrun error occurs, the previous data (which is still present in the TX registers that were not written) is retransmitted. If TEIE is set, an ESAI transmit data with exception (underrun error) interrupt request is issued when TUE is set. Hardware, software, ESAI individual, and STOP reset clear TUE. TUE is also cleared by reading the SAISR with TUE set, followed by writing to all the enabled transmit data registers or to TSR.

8.3.6.12 SAISR Transmit Data Register Empty (TDE) - Bit 15

TDE is set when the contents of the transmit data register of all the enabled transmitters are transferred to the transmit shift registers; it is also set for a TSR disabled time slot period in network mode (as if data were being transmitted after the TSR was written). When set, TDE indicates that data should be written to all the TX registers of the enabled transmitters or to the time slot register (TSR). TDE is cleared when the DSP writes to all the transmit data registers of the enabled transmitters, or when the DSP writes to the TSR to disable transmission of the next time slot. If TIE is set, an ESAI transmit data interrupt request is issued when TDE is set. Hardware, software, ESAI individual, and STOP reset clear TDE.

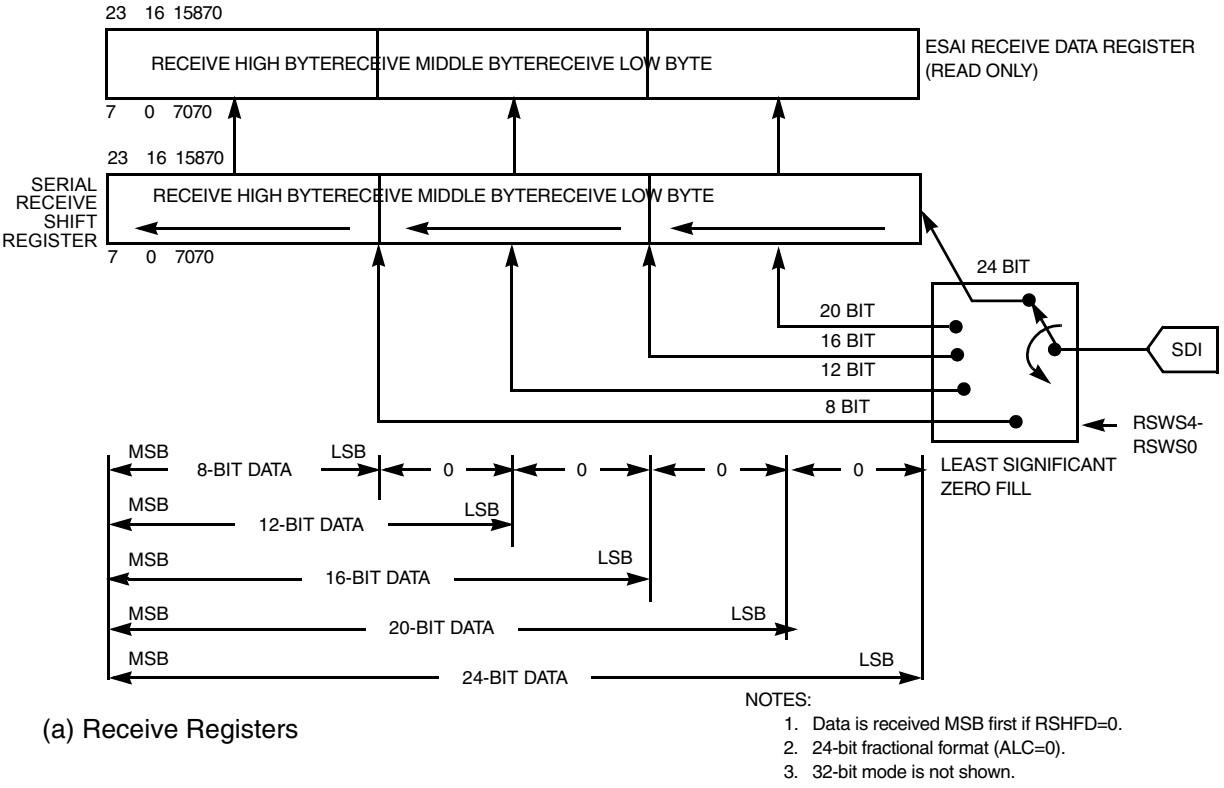
8.3.6.13 SAISR Transmit Even-Data Register Empty (TEDE) - Bit 16

When set, TEDE indicates that the enabled transmitter data registers became empty at the beginning of an even time slot. Even time slots are all even-numbered slots (0, 2, 4, 6, etc.). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. The zero time slot is considered even. This flag is set when the contents of the transmit data register of all the enabled transmitters are transferred to the transmit shift registers; it is also set for a TSR disabled time slot period in network mode (as if data were being transmitted after the TSR was written). When set, TEDE indicates that data should be written to all the TX registers of the enabled transmitters or to the time slot register (TSR). TEDE is cleared when the DSP writes to all the transmit data registers of the enabled transmitters, or when the DSP writes to the TSR to disable transmission of the next time slot. If TIE is set, an ESAI transmit data interrupt request is issued when TEDE is set. Hardware, software, ESAI individual, and STOP reset clear TEDE.

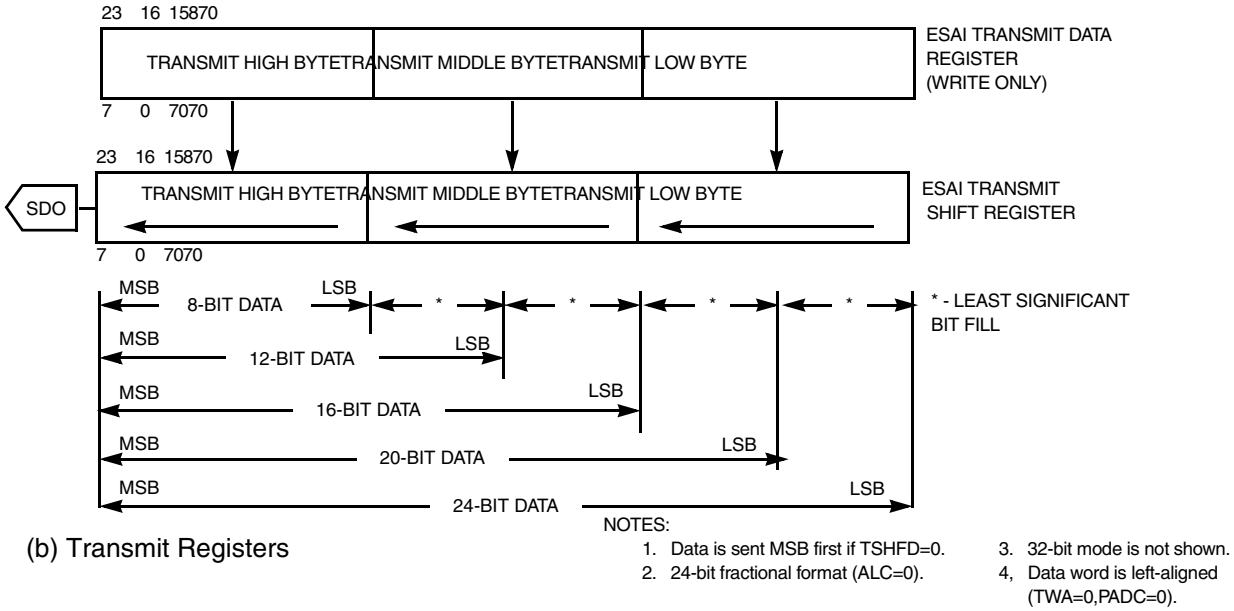
8.3.6.14 SAISR Transmit Odd-Data Register Empty (TODE) - Bit 17

When set, TODE indicates that the enabled transmitter data registers became empty at the beginning of an odd time slot. Odd time slots are all odd-numbered slots (1, 3, 5, etc.). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. This flag is set when the contents of the transmit data register of all the enabled transmitters are transferred to the transmit shift registers; it is also set for a

TSR disabled time slot period in network mode (as if data were being transmitted after the TSR was written). When set, TODE indicates that data should be written to all the TX registers of the enabled transmitters or to the time slot register (TSR). TODE is cleared when the DSP writes to all the transmit data registers of the enabled transmitters, or when the DSP writes to the TSR to disable transmission of the next time slot. If TIE is set, an ESAI transmit data interrupt request is issued when TODE is set. Hardware, software, ESAI individual, and STOP reset clear TODE.

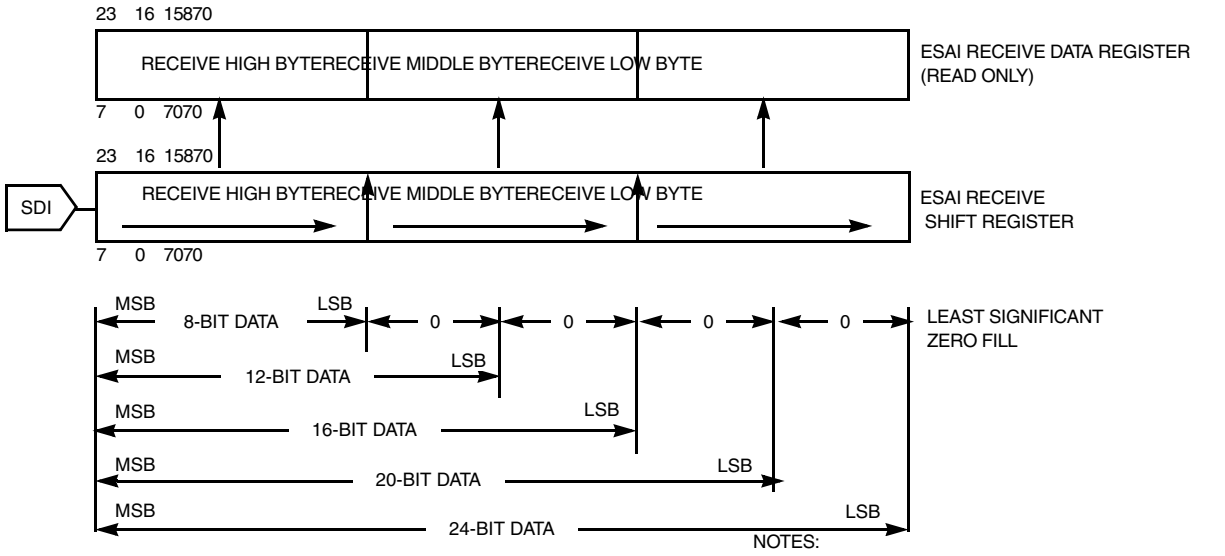


(a) Receive Registers



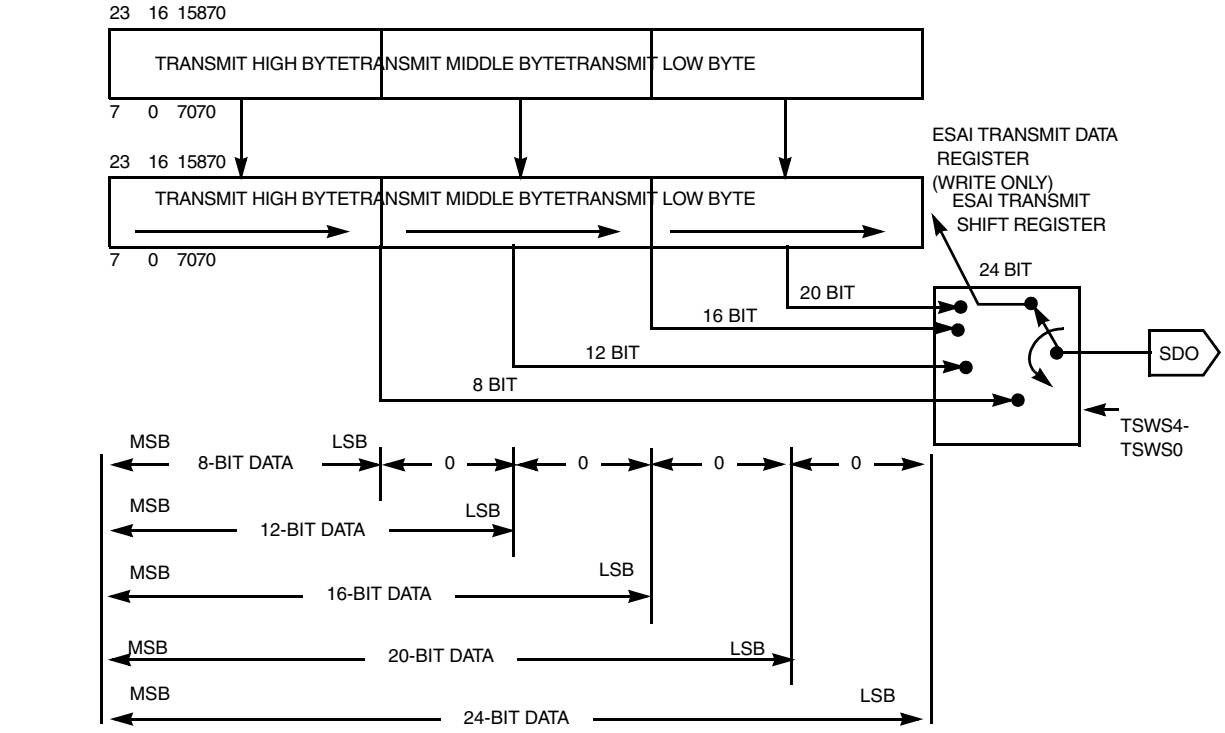
(b) Transmit Registers

Figure 8-13 ESAI Data Path Programming Model ([R/T]SHFD=0)



(a) Receive Registers

NOTES:
1. Data is received LSB first if RSHFD=1.
2. 24-bit fractional format (ALC=0).
3. 32-bit mode is not shown.



(b) Transmit Registers

NOTES:
1. Data is sent LSB first if TSHFD=1.
2. 24-bit fractional format (ALC=0).
3. 32-bit mode is not shown.
4. Data word is left aligned (TWA=0,PADC=1).

Figure 8-14 ESAI Data Path Programming Model ([R/T]SHFD=1)

8.3.7 ESAI Receive Shift Registers

The receive shift registers (see Figure 8-13 and Figure 8-14) receive the incoming data from the serial receive data pins. Data is shifted in by the selected (internal/external) bit clock when the associated frame sync I/O is asserted. Data is assumed to be received MSB first if RSHFD=0 and LSB first if RSHFD=1. Data is transferred to the ESAI receive data registers after 8, 12, 16, 20, 24, or 32 serial clock cycles were counted, depending on the slot length control bits in the RCR register.

8.3.8 ESAI Receive Data Registers (RX3, RX2, RX1, RX0)

RX3, RX2, RX1 and RX0 are 24-bit read-only registers that accept data from the receive shift registers when they become full (see Figure 8-13 and Figure 8-14). The data occupies the most significant portion of the receive data registers, according to the ALC control bit setting. The unused bits (least significant portion, and 8 most significant bits when ALC=1) read as zeros. The DSP is interrupted whenever RXx becomes full if the associated interrupt is enabled.

8.3.9 ESAI Transmit Shift Registers

The transmit shift registers contain the data being transmitted (see Figure 8-13 and Figure 8-14). Data is shifted out to the serial transmit data pins by the selected (internal/external) bit clock when the associated frame sync I/O is asserted. The number of bits shifted out before the shift registers are considered empty and may be written to again can be 8, 12, 16, 20, 24 or 32 bits (determined by the slot length control bits in the TCR register). Data is shifted out of these registers MSB first if TSHFD=0 and LSB first if TSHFD=1.

8.3.10 ESAI Transmit Data Registers (TX5, TX4, TX3, TX2,TX1,TX0)

TX5, TX4, TX3, TX2, TX1 and TX0 are 24-bit write-only registers. Data to be transmitted is written into these registers and is automatically transferred to the transmit shift registers (see Figure 8-13 and Figure 8-14). The data written (8, 12, 16, 20 or 24 bits) should occupy the most significant portion of the TXx according to the ALC control bit setting. The unused bits (least significant portion, and the 8 most significant bits when ALC=1) of the TXx are don’t care bits. The DSP is interrupted whenever the TXx becomes empty if the transmit data register empty interrupt has been enabled.

8.3.11 ESAI Time Slot Register (TSR)

The write-only Time Slot Register (TSR) is effectively a null data register that is used when the data is not to be transmitted in the available transmit time slot. The transmit data pins of all the enabled transmitters are in the high-impedance state for the respective time slot where TSR has been written. The Transmitter External Buffer Enable pin (FSR pin when SYN=1, TEBE=1, RFSD=1) disables the external buffers during the slot when the TSR register has been written.

8.3.12 Transmit Slot Mask Registers (TSMA, TSMB)

The Transmit Slot Mask Registers (TSMA and TSMB) are two read/write registers used by the transmitters in network mode to determine for each slot whether to transmit a data word and generate a

transmitter empty condition (TDE=1), or to tri-state the transmitter data pins. TSMA and TSMB should each be considered as containing half a 32-bit register TSM. See Figure 8-15 and Figure 8-16. Bit number N in TSM (TS**) is the enable/disable control bit for transmission in slot number N.

	11	10	9	8	7	6	5	4	3	2	1	0
X:$FFFFB9	TS11	TS10	TS9	TS8	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0

	23	22	21	20	19	18	17	16	15	14	13	12
									TS15	TS14	TS13	TS12

Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 8-15 TSMA Register

	11	10	9	8	7	6	5	4	3	2	1	0
X:$FFFFBA	TS27	TS26	TS25	TS24	TS23	TS22	TS21	TS20	TS19	TS18	TS17	TS16

	23	22	21	20	19	18	17	16	15	14	13	12
									TS31	TS30	TS29	TS28

Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 8-16 TSMB Register

When bit number N in TSM is cleared, all the transmit data pins of the enabled transmitters are tri-stated during transmit time slot number N. The data is still transferred from the transmit data registers to the transmit shift registers but neither the TDE nor the TUE flags are set. This means that during a disabled slot, no transmitter empty interrupt is generated. The DSP is interrupted only for enabled slots. Data that is written to the transmit data registers when servicing this request is transmitted in the next enabled transmit time slot.

When bit number N in TSM register is set, the transmit sequence is as usual: data is transferred from the TX registers to the shift registers, transmitted during slot number N, and the TDE flag is set.

Using the slot mask in TSM does not conflict with using TSR. Even if a slot is enabled in TSM, the user may chose to write to TSR instead of writing to the transmit data registers TXx. This causes all the transmit data pins of the enabled transmitters to be tri-stated during the next slot.

Data written to the TSM affects the next frame transmission. The frame being transmitted is not affected by this data and would comply to the last TSM setting. Data read from TSM returns the last written data.

After hardware or software reset, the TSM register is preset to $FFFFFFFF, which means that all 32 possible slots are enabled for data transmission.

NOTE

When operating in normal mode, bit 0 of the mask register must be set, otherwise no output is generated.

8.3.13 Receive Slot Mask Registers (RSMA, RSMB)

The Receive Slot Mask Registers (RSMA and RSMB) are two read/write registers used by the receiver in network mode to determine for each slot whether to receive a data word and generate a receiver full condition (RDF=1), or to ignore the received data. RSMA and RSMB should be considered as each containing half of a 32-bit register RSM. See Figure 8-17 and Figure 8-18. Bit number N in RSM (RS**) is an enable/disable control bit for receiving data in slot number N.

	11	10	9	8	7	6	5	4	3	2	1	0
X:$FFFFBB	RS11	RS10	RS9	RS8	RS7	RS6	RS5	RS4	RS3	RS2	RS1	RS0

	23	22	21	20	19	18	17	16	15	14	13	12
									RS15	RS14	RS13	RS12

Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 8-17 RSMA Register

	11	10	9	8	7	6	5	4	3	2	1	0
X:$FFFFBC	RS27	RS26	RS25	RS24	RS23	RS22	RS21	RS20	RS19	RS18	RS17	RS16

	23	22	21	20	19	18	17	16	15	14	13	12
									RS31	RS30	RS29	RS28

Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 8-18 RSMB Register

When bit number N in the RSM register is cleared, the data from the enabled receivers input pins are shifted into their receive shift registers during slot number N. The data is not transferred from the receive shift registers to the receive data registers, and neither the RDF nor the ROE flags are set. This means that during a disabled slot, no receiver full interrupt is generated. The DSP is interrupted only for enabled slots.

When bit number N in the RSM is set, the receive sequence is as usual: data which is shifted into the enabled receivers shift registers is transferred to the receive data registers and the RDF flag is set.

Data written to the RSM affects the next received frame. The frame being received is not affected by this data and would comply to the last RSM setting. Data read from RSM returns the last written data.

After hardware or software reset, the RSM register is preset to $FFFFFFFF, which means that all 32 possible slots are enabled for data reception.

NOTE

When operating in normal mode, bit 0 of the mask register must be set to one, otherwise no input is received.

8.4 Operating Modes

ESAI operating mode are selected by the ESAI control registers (TCCR, TCR, RCCR, RCR and SAICR). The main operating mode are described in the following paragraphs.

8.4.1 ESAI After Reset

Hardware or software reset clears the port control register bits and the port direction control register bits, which configure all ESAI I/O pins as disconnected. The ESAI is in the individual reset state while all ESAI pins are programmed as GPIO or disconnected, and is active only if at least one of the ESAI I/O pins is programmed as an ESAI pin.

8.4.2 ESAI Initialization

The correct way to initialize the ESAI is as follows:

1. Hardware, software, ESAI individual, or STOP reset.
2. Program ESAI control and time slot registers.
3. Write data to all the enabled transmitters.
4. Configure at least one pin as ESAI pin.

During program execution, all ESAI pins may be defined as GPIO or disconnected, causing the ESAI to stop serial activity and enter the individual reset state. All status bits of the interface are set to their reset state; however, the control bits are not affected. This procedure allows the DSP programmer to reset the ESAI separately from the other internal peripherals. During individual reset, internal DMA accesses to the data registers of the ESAI are not valid and data read is undefined.

The DSP programmer must use an individual ESAI reset when changing the ESAI control registers (except for TEIE, REIE, TLIE, RLIE, TIE, RIE, TE0-TE5, RE0-RE3) to ensure proper operation of the interface.

NOTE

If the ESAI receiver section is already operating with some of the receivers, enabling additional receivers on the fly (i.e., without first putting the ESAI receiver in the personal reset state) by setting their REx control bits will result in erroneous data being received as the first data word for the newly enabled receivers.

8.4.3 ESAI Interrupt Requests

The ESAI can generate eight different interrupt requests (ordered from the highest to the lowest priority)**:**

1. ESAI Receive Data with Exception Status.
   Occurs when the receive exception interrupt is enabled (REIE=1 in the RCR register), at least one of the enabled receive data registers is full (RDF=1), and a receiver overrun error has occurred (ROE=1 in the SAISR register). ROE is cleared by first reading the SAISR and then reading all the enabled receive data registers.
2. ESAI Receive Even Data
   Occurs when the receive even slot data interrupt is enabled (REDIE=1), at least one of the enabled receive data registers is full (RDF=1), the data is from an even slot (REDF=1), and no exception has occurred (ROE=0 or REIE=0).
   Reading all enabled receiver data registers clears RDF and REDF.
3. ESAI Receive Data
   Occurs when the receive interrupt is enabled (RIE=1), at least one of the enabled receive data registers is full (RDF=1), no exception has occurred (ROE=0 or REIE=0), and no even slot interrupt has occurred (REDF=0 or REDIE=0).
   Reading all enabled receiver data registers clears RDF.
4. ESAI Receive Last Slot Interrupt
   Occurs, if enabled (RLIE=1), after the last slot of the frame ended (in network mode only) regardless of the receive mask register setting. The receive last slot interrupt may be used for resetting the receive mask slot register, reconfiguring the DMA channels and reassigning data memory pointers. Using the receive last slot interrupt guarantees that the previous frame was serviced with the previous setting and the new frame is serviced with the new setting without synchronization problems. Note that the maximum receive last slot interrupt service time should not exceed N-1 ESAI bits service time (where N is the number of bits in a slot).
5. ESAI Transmit Data with Exception Status
   Occurs when the transmit exception interrupt is enabled (TEIE=1), at least one transmit data register of the enabled transmitters is empty (TDE=1), and a transmitter underrun error has occurred (TUE=1). TUE is cleared by first reading the SAISR and then writing to all the enabled transmit data registers, or to the TSR register.
6. ESAI Transmit Last Slot Interrupt
   Occurs, if enabled (TLIE=1), at the start of the last slot of the frame in network mode regardless of the transmit mask register setting. The transmit last slot interrupt may be used for resetting the transmit mask slot register, reconfiguring the DMA channels and reassigning data memory pointers. Using the transmit last slot interrupt guarantees that the previous frame was serviced with the previous setting and the new frame is serviced with the new setting without synchronization problems. Note that the maximum transmit last slot interrupt service time should not exceed N-1 ESAI bits service time (where N is the number of bits in a slot).
7. ESAI Transmit Even Data
   Occurs when the transmit even slot data interrupt is enabled (TEDIE=1), at least one of the enabled transmit data registers is empty (TDE=1), the slot is an even slot (TEDE=1), and no exception has occurred (TUE=0 or TEIE=0).
   Writing to all the TX registers of the enabled transmitters or to TSR clears this interrupt request.

8. ESAI Transmit Data
   Occurs when the transmit interrupt is enabled (TIE=1), at least one of the enabled transmit data registers is empty (TDE=1), no exception has occurred (TUE=0 or TEIE=0), and no even slot interrupt has occurred (TEDE=0 or TEDIE=0).
   Writing to all the TX registers of the enabled transmitters, or to the TSR clears this interrupt request.

8.4.4 Operating Modes – Normal, Network, and On-Demand

The ESAI has three basic operating modes and many data/operation formats.

8.4.4.1 Normal/Network/On-Demand Mode Selection

Selecting between the normal mode and network mode is accomplished by clearing or setting the TMOD0-TMOD1 bits in the TCR register for the transmitter section, and in the RMOD0-RMOD1 bits in the RCR register for the receiver section.

For normal mode, the ESAI functions with one data word of I/O per frame (per enabled transmitter or receiver). The normal mode is typically used to transfer data to/from a single device.

For the network mode, 2 to 32 time slots per frame may be selected. During each frame, 0 to 32 data words of I/O may be received/transmitted. In either case, the transfers are periodic. The frame sync signal indicates the first time slot in the frame. Network mode is typically used in time division multiplexed (TDM) networks of codecs, DSPs with multiple words per frame, or multi-channel devices.

Selecting the network mode and setting the frame rate divider to zero (DC=00000) selects the on-demand mode. This special case does not generate a periodic frame sync. A frame sync pulse is generated only when data is available to transmit. The on-demand mode requires that the transmit frame sync be internal (output) and the receive frame sync be external (input). Therefore, for simplex operation, the synchronous mode could be used; however, for full-duplex operation, the asynchronous mode must be used. Data transmission that is data driven is enabled by writing data into each TX. Although the ESAI is double buffered, only one word can be written to each TX, even if the transmit shift register is empty. The receive and transmit interrupts function as usual using TDE and RDF; however, transmit underruns are impossible for on-demand transmission and are disabled.

8.4.4.2 Synchronous/Asynchronous Operating Modes

The transmit and receive sections of the ESAI may be synchronous or asynchronous – i.e., the transmitter and receiver sections may use common clock and synchronization signals (synchronous operating mode), or they may have their own separate clock and sync signals (asynchronous operating mode). The SYN bit in the SAICR register selects synchronous or asynchronous operation. Since the ESAI is designed to operate either synchronously or asynchronously, separate receive and transmit interrupts are provided.

When SYN is cleared, the ESAI transmitter and receiver clocks and frame sync sources are independent. If SYN is set, the ESAI transmitter and receiver clocks and frame sync come from the transmitter section (either external or internal sources).

Data clock and frame sync signals can be generated internally by the DSP or may be obtained from external sources. If internally generated, the ESAI clock generator is used to derive high frequency clock, bit clock and frame sync signals from the DSP internal system clock.

8.4.4.3 Frame Sync Selection

The frame sync can be either a bit-long or word-long signal. The transmitter frame format is defined by the TFSL bit in the TCR register. The receiver frame format is defined by the RFSL bit in the RCR register.

1. In the word-long frame sync format, the frame sync signal is asserted during the entire word data transfer period. This frame sync length is compatible with Freescale codecs, SPI serial peripherals, serial A/D and D/A converters, shift registers, and telecommunication PCM serial I/O.
2. In the bit-long frame sync format, the frame sync signal is asserted for one bit clock immediately before the data transfer period. This frame sync length is compatible with Intel and National components, codecs, and telecommunication PCM serial I/O.

The relative timing of the word length frame sync as referred to the data word is specified by the TFSR bit in the TCR register for the transmitter section, and by the RFSR bit in the RCR register for the receive section. The word length frame sync may be generated (or expected) with the first bit of the data word, or with the last bit of the previous word. TFSR and RFSR are ignored when a bit length frame sync is selected.

Polarity of the frame sync signal may be defined as positive (asserted high) or negative (asserted low). The TFSP bit in the TCCR register specifies the polarity of the frame sync for the transmitter section. The RFSP bit in the RCCR register specifies the polarity of the frame sync for the receiver section.

The ESAI receiver looks for a receive frame sync leading edge (trailing edge if RFSP is set) only when the previous frame is completed. If the frame sync goes high before the frame is completed (or before the last bit of the frame is received in the case of a bit frame sync or a word length frame sync with RFSR set), the current frame sync is not recognized, and the receiver is internally disabled until the next frame sync. Frames do not have to be adjacent – i.e., a new frame sync does not have to immediately follow the previous frame. Gaps of arbitrary periods can occur between frames. Enabled transmitters are tri-stated during these gaps.

When operating in the synchronous mode (SYN=1), all clocks including the frame sync are generated by the transmitter section.

8.4.4.4 Shift Direction Selection

Some data formats, such as those used by codecs, specify MSB first while other data formats, such as the AES-EBU digital audio interface, specify LSB first. The MSB/LSB first selection is made by programming RSHFD bit in the RCR register for the receiver section, and by programming the TSHFD bit in the TCR register for the transmitter section.

8.4.5 Serial I/O Flags

Three ESAI pins (FSR, SCKR and HCKR) are available as serial I/O flags when the ESAI is operating in the synchronous mode (SYN=1). Their operation is controlled by RCKD, RFSD, TEBE bits in the RCR,

RCCR and SAICR registers.The output data bits (OF2, OF1 and OF0) and the input data bits (IF2, IF1 and IF0) are double buffered to/from the HCKR, FSR and SCKR pins. Double buffering the flags keeps them in sync with the TX and RX data lines.

Each flag can be separately programmed. Flag 0 (SCKR pin) direction is selected by RCKD, RCKD=1 for output and RCKD=0 for input. Flag 1 (FSR pin) is enabled when the pin is not configured as external transmitter buffer enable (TEBE=0) and its direction is selected by RFSD, RFSD=1 for output and RFSD=0 for input. Flag 2 (HCKR pin) direction is selected by RHCKD, RHCKD=1 for output and RHCKD=0 for input.

When programmed as input flags, the SCKR, FSR and HCKR logic values, respectively, are latched at the same time as the first bit of the receive data word is sampled. Because the input was latched, the signal on the input flag pin (SCKR, FSR or HCKR) can change without affecting the input flag until the first bit of the next receive data word. When the received data words are transferred to the receive data registers, the input flag latched values are then transferred to the IF0, IF1 and IF2 bits in the SAISR register, where they may be read by software.

When programmed as output flags, the SCKR, FSR and HCKR logic values are driven by the contents of the OF0, OF1 and OF2 bits in the SAICR register respectively, and are driven when the transmit data registers are transferred to the transmit shift registers. The value on SCKR, FSR and HCKR is stable from the time the first bit of the transmit data word is transmitted until the first bit of the next transmit data word is transmitted. Software may change the OF0-OF2 values thus controlling the SCKR, FSR and HCKR pin values for each transmitted word. The normal sequence for setting output flags when transmitting data is as follows: wait for TDE (transmitter empty) to be set, first write the flags, and then write the transmit data to the transmit registers. OF0, OF1 and OF2 are double buffered so that the flag states appear on the pins when the transmit data is transferred to the transmit shift register (i.e., the flags are synchronous with the data).

8.5 GPIO - Pins and Registers

The GPIO functionality of the ESAI port is controlled by three registers: Port C control register (PCRC), Port C direction register (PRRC) and Port C data register (PDRC).

8.5.1 Port C Control Register (PCRC)

The read/write 24-bit Port C Control Register (PCRC) in conjunction with the Port C Direction Register (PRRC) controls the functionality of the ESAI GPIO pins. Each of the PC(11:0) bits controls the functionality of the corresponding port pin. See Table 8-12 for the port pin configurations. Hardware and software reset clear all PCRC bits.

8.5.2 Port C Direction Register (PRRC)

The read/write 24-bit Port C Direction Register (PRRC) in conjunction with the Port C Control Register (PCRC) controls the functionality of the ESAI GPIO pins. Table 8-12 describes the port pin configurations. Hardware and software reset clear all PRRC bits.

Table 8-12 PCRC and PRRC Bits Functionality

PDC[i]	PC[i]	Port Pin[i] Function
0	0	disconnected
0	1	GPIO input
1	0	GPIO output
1	1	ESAI

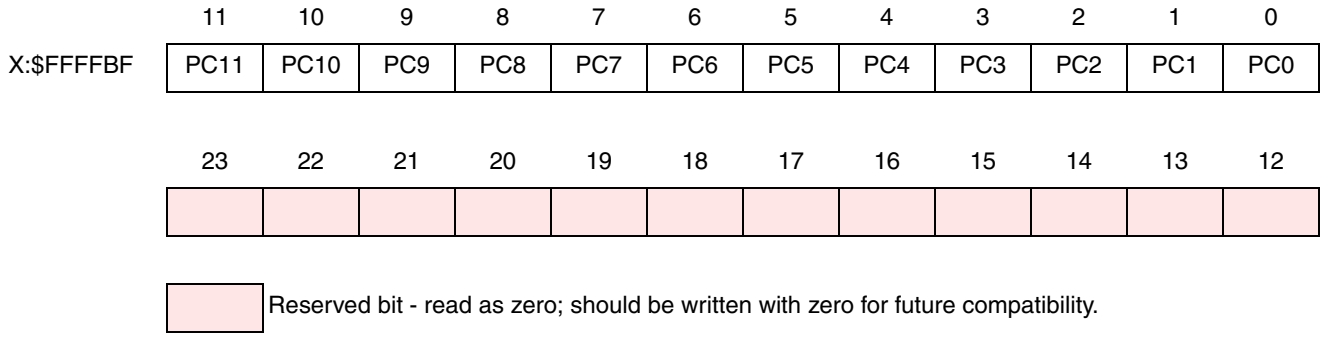

11 10 9 8 7 6 5 4 3 2 1 0
X:$FFFFBF
PC11 PC10 PC9 PC8 PC7 PC6 PC5 PC4 PC3 PC2 PC1 PC0
23 22 21 20 19 18 17 16 15 14 13 12
Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 8-19 PCRC Register

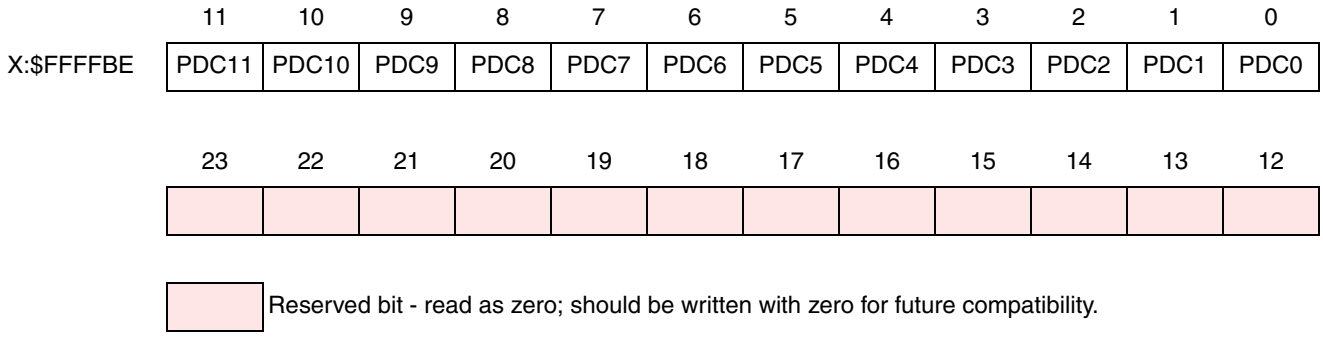

11 10 9 8 7 6 5 4 3 2 1 0
X:$FFFFBE
PDC11 PDC10 PDC9 PDC8 PDC7 PDC6 PDC5 PDC4 PDC3 PDC2 PDC1 PDC0
23 22 21 20 19 18 17 16 15 14 13 12
Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 8-20 PRRC Register

8.5.3 Port C Data register (PDRC)

The read/write 24-bit Port C Data Register (see Figure 8-21) is used to read or write data to/from ESAI GPIO pins. Bits PD(11:0) are used to read or write data from/to the corresponding port pins if they are configured as GPIO. If a port pin [i] is configured as a GPIO input, then the corresponding PD[i] bit reflects the value present on this pin. If a port pin [i] is configured as a GPIO output, then the value written into the corresponding PD[i] bit is reflected on this pin. If a port pin [i] is configured as disconnected, the corresponding PD[i] bit is not reset and contains undefined data.

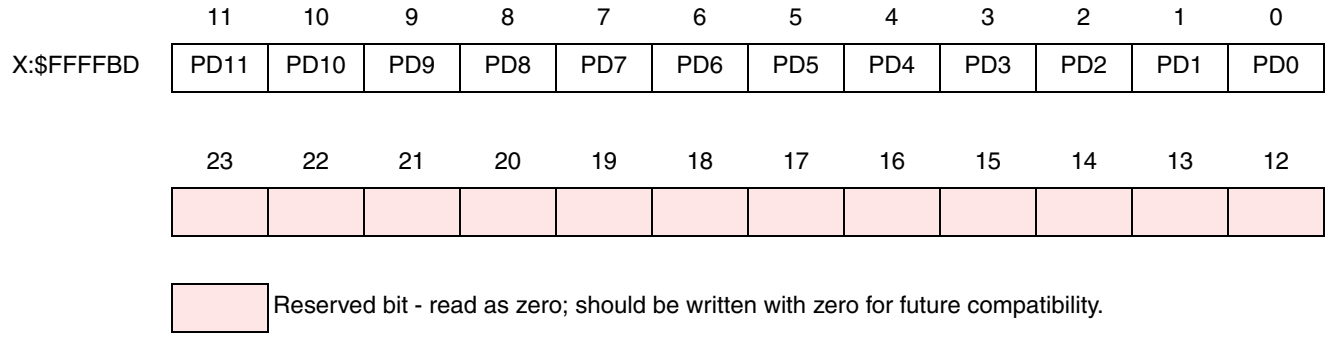

11 10 9 8 7 6 5 4 3 2 1 0
X:$FFFFBD
PD11 PD10 PD9 PD8 PD7 PD6 PD5 PD4 PD3 PD2 PD1 PD0
23 22 21 20 19 18 17 16 15 14 13 12
Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 8-21 PDRC Register

8.6 ESAI Initialization Examples

8.6.1 Initializing the ESAI Using Individual Reset

- The ESAI should be in its individual reset state (PCRC = $000 and PRRC = $000). In the individual reset state, both the transmitter and receiver sections of the ESAI are simultaneously reset. The TPR bit in the TCR register may be used to reset just the transmitter section. The RPR bit in the RCR register may be used to reset just the receiver section.
- Configure the control registers (TCCR, TCR, RCCR, RCR) according to the operating mode, but do not enable transmitters (TE5–TE0 = $0) or receivers (RE3–RE0 = $0). It is possible to set the interrupt enable bits which are in use during the operation (no interrupt occurs).
- Enable the ESAI by setting the PCRC register and PRRC register bits according to pins which are in use during operation.
- Write the first data to be transmitted to the transmitters which are in use during operation. This step is needed even if DMA is used to service the transmitters.
- Enable the transmitters and receivers.
- From now on ESAI can be serviced either by polling, interrupts, or DMA.

Operation proceeds as follows:

- For internally generated clock and frame sync, these signals are active immediately after ESAI is enabled (step 3 above).
- Data is received only when one of the receive enable (REx) bits is set and after the occurrence of frame sync signal (either internally or externally generated).
- Data is transmitted only when the transmitter enable (TEx) bit is set and after the occurrence of frame sync signal (either internally or externally generated). The transmitter outputs remain tri-stated after TEx bit is set until the frame sync occurs.

8.6.2 Initializing Just the ESAI Transmitter Section

- It is assumed that the ESAI is operational; that is, at least one pin is defined as an ESAI pin.
- The transmitter section should be in its personal reset state (TPR = 1).

- Configure the control registers TCCR and TCR according to the operating mode, making sure to clear the transmitter enable bits (TE0 - TE5). TPR must remain set.
- Take the transmitter section out of the personal reset state by clearing TPR.
- Write first data to the transmitters which will be used during operation. This step is needed even if DMA is used to service the transmitters.
- Enable the transmitters by setting their TE bits.
- Data is transmitted only when the transmitter enable (TEx) bit is set and after the occurrence of frame sync signal (either internally or externally generated). The transmitter outputs remain tri-stated after TEx bit is set until the frame sync occurs.
- From now on the transmitters are operating and can be serviced either by polling, interrupts, or DMA.

8.6.3 Initializing Just the ESAI Receiver Section

- It is assumed that the ESAI is operational; that is, at least one pin is defined as an ESAI pin.
- The receiver section should be in its personal reset state (RPR = 1).
- Configure the control registers RCCR and RCR according to the operating mode, making sure to clear the receiver enable bits (RE0 - RE3). RPR must remain set.
- Take the receiver section out of the personal reset state by clearing RPR.
- Enable the receivers by setting their RE bits.
- From now on the receivers are operating and can be serviced either by polling, interrupts, or DMA.

9 Enhanced Serial Audio Interface 1 (ESAI_1)

9.1 Introduction

The Enhanced Serial Audio Interface I (ESAI_1) is the second ESAI peripheral in the DSP56366. It is functionally identical to the ESAI peripheral described in Section 8, "Enhanced Serial AUDIO Interface (ESAI)" except for minor differences described in this section. Refer to the ESAI section for functional information about the ESAI_1, in addition to using the information in this section.

The ESAI_1 block diagram is shown in Figure 9-1. The ESAI_1 shares 4 pins with the ESAI. The ESAI_1 does not have the two high frequency clock pins but otherwise it is identical to the ESAI.

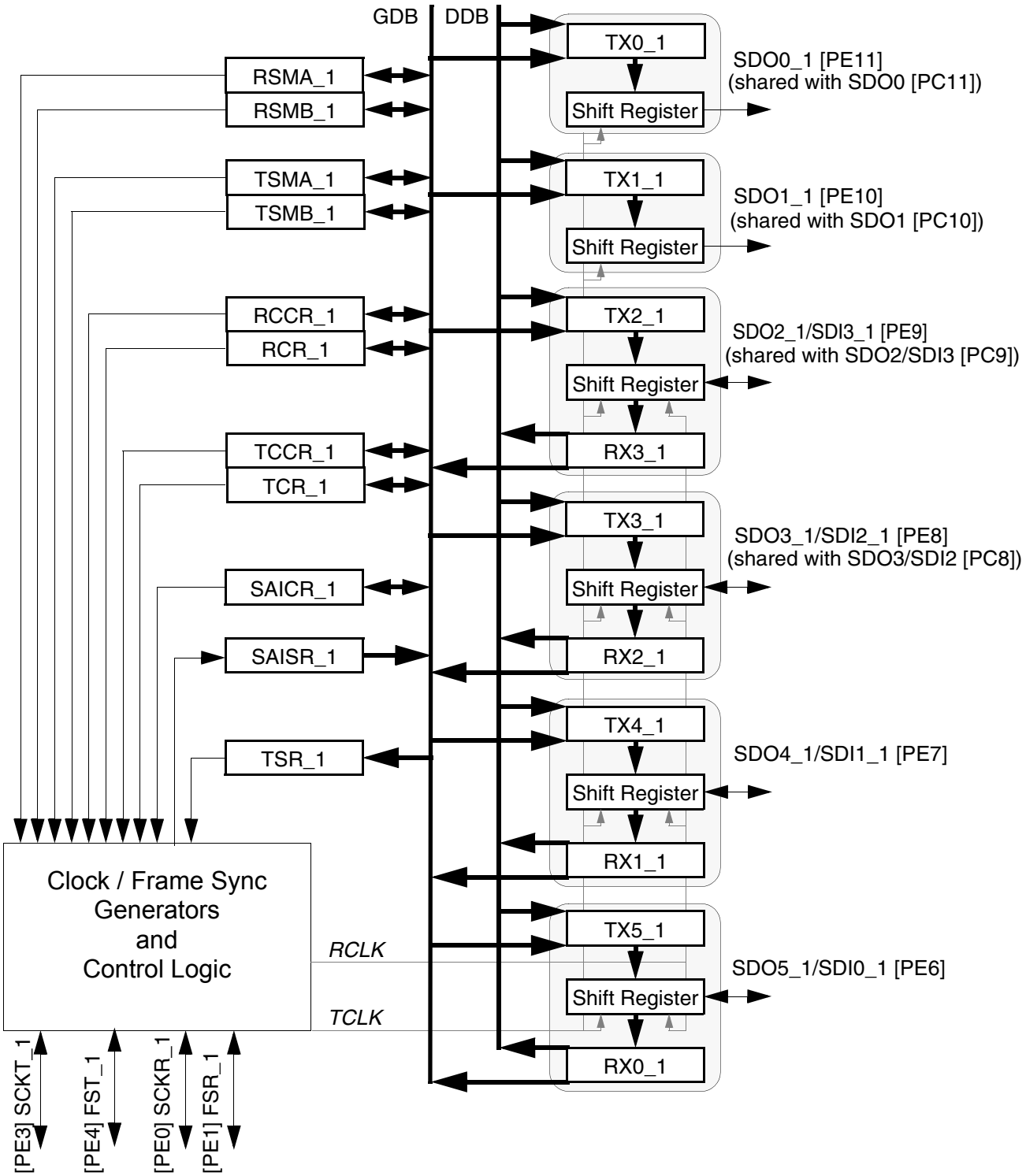


Figure 9-1 ESAI_1 Block Diagram

9.2 ESAI_1 Data and Control Pins

The ESAI_1 has 6 dedicated pins and shares 4 pins with the ESAI. The pins are described in the following sections.

9.2.1 Serial Transmit 0 Data Pin (SDO0_1)

SDO0_1 transmits data from the TX0_1 serial transmit shift register. It is shared with the ESAI SDO0 signal. The pin may be used as SDO0_1 if it is not defined as ESAI SDO0. The pin may be used as GPIO PE11 if not used by the ESAI or ESAI_1. The ESAI_1 Multiplex Control Register (EMUXR) defines if the pin belongs to the ESAI or to the ESAI_1.

9.2.2 Serial Transmit 1 Data Pin (SDO1_1)

SDO1_1 transmits data from the TX1_1 serial transmit shift register. It is shared with the ESAI SDO1 signal. The pin may be used as SDO1_1 if it is not defined as ESAI SDO1. The pin may be used as GPIO PE10 if not used by the ESAI or ESAI_1. The ESAI_1 Multiplex Control Register (EMUXR) defines if the pin belongs to the ESAI or to the ESAI_1.

9.2.3 Serial Transmit 2/Receive 3 Data Pin (SDO2_1/SDI3_1)

SDO2_1/SDI3_1 transmits data from the TX2_1 serial transmit shift register when programmed as a transmitter pin, or receives serial data to the RX3_1 serial receive shift register when programmed as a receiver pin. It is shared with the ESAI SDO2/SDI3 signal. The pin may be used as SDO2_1/SDI3_1 if it is not defined as ESA I SDO2/SDI3. The pin may be used as GPIO PE9 if not used by the ESAI or ESAI_1. The ESAI_1 Multiplex Control Register (EMUXR) defines if the pin belongs to the ESAI or to the ESAI_1.

9.2.4 Serial Transmit 3/Receive 2 Data Pin (SDO3_1/SDI2_1)

SDO3_1/SDI2_1 transmits data from the TX3_1 serial transmit shift register when programmed as a transmitter pin, or receives serial data to the RX2_1 serial receive shift register when programmed as a receiver pin. It is shared with the ESAI SDO3/SDI2 signal. The pin may be used as SDO3_1/SDI2_1 if it is not defined as ESAI SDO3/SDI2. The pin may be used as GPIO PE8 if not used by the ESAI or ESAI_1. The ESAI_1 Multiplex Control Register (EMUXR) defines if the pin belongs to the ESAI or to the ESAI_1.

9.2.5 Serial Transmit 4/Receive 1 Data Pin (SDO4_1/SDI1_1)

SDO4_1/SDI1_1 transmits data from the TX4_1 serial transmit shift register when programmed as a transmitter pin, or receives serial data to the RX1_1 serial receive shift register when programmed as a receiver pin. SDO4_1/SDI1_1 may be programmed as a general-purpose pin (PE7) when the ESAI_1 SDO4_1 and SDI1_1 functions are not being used.

9.2.6 Serial Transmit 5/Receive 0 Data Pin (SDO5_1/SDI0_1)

SDO5_1/SDI0_1 transmits data from the TX5_1 serial transmit shift register when programmed as transmitter pin, or receives serial data to the RX0_1 serial shift register when programmed as a receiver pin. SDO5_1/SDI0_1 may be programmed as a general-purpose pin (PE6) when the ESAI_1 SDO5_1 and SDI0_1 functions are not being used.

9.2.7 Receiver Serial Clock (SCKR_1)

SCKR_1 is a bidirectional pin that provides the receivers serial bit clock for the ESAI_1 interface. SCKR_1 may be programmed as a general-purpose I/O pin (PE0) when the ESAI_1 SCKR_1 function is not being used.

9.2.8 Transmitter Serial Clock (SCKT_1)

SCKT_1 is a bidirectional pin that provides the transmitters serial bit clock for the ESAI_1 interface. SCKT_1 may be programmed as a general-purpose I/O pin (PE3) when the ESAI_1 SCKT_1 function is not being used.

9.2.9 Frame Sync for Receiver (FSR_1)

The FSR_1 pin is a bidirectional pin that provides the receivers frame sync signal for the ESAI_1 interface. FSR_1 may be programmed as a general-purpose I/O pin (PE1) when the ESAI_1 FSR_1 function is not being used.

9.2.10 Frame Sync for Transmitter (FST_1)

The FST_1 pin is a bidirectional pin that provides the transmitters frame sync signal for the ESAI_1 interface. FST_1 may be programmed as a general-purpose I/O pin (PE4) when the ESAI_1 FST_1 function is not being used.

9.3 ESAI_1 Programming Model

The ESAI_1 has the following registers:

- One multiplex control register
- Five control registers
- One status register
- Six transmit data registers
- Four receive data registers
- Two transmit slot mask registers
- Two receive slot mask registers
- One special-purpose time slot register

The ESAI_1 also contains the GPIO Port E functionality, described in Section 9.5, "GPIO - Pins and Registers". The following paragraphs give detailed descriptions of bits in the ESAI_1 registers that differ in functionality from their descriptions in the ESAI Programming Model.

9.3.1 ESAI_1 Multiplex Control Register (EMUXR)

The read/write ESAI_1 Multiplex Control Register (EMUXR) controls which peripheral (ESAI or ESAI_1) is using the shared pins.

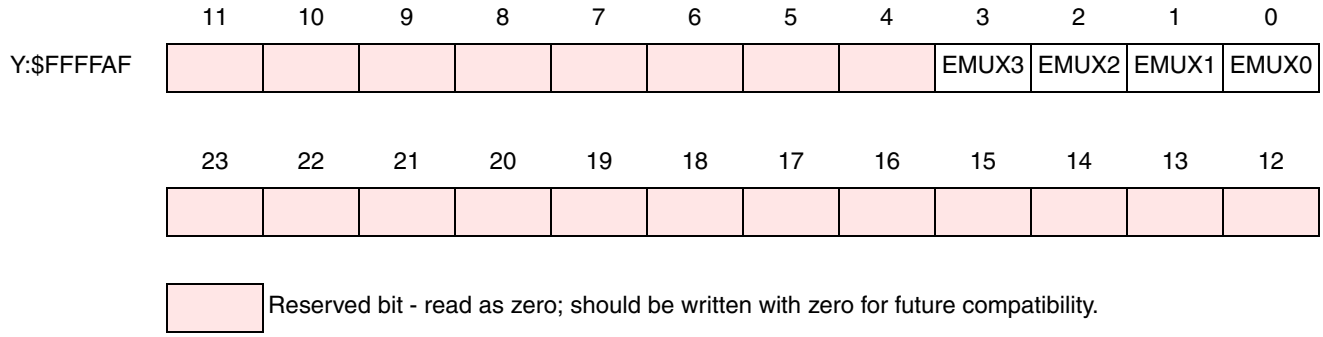

11 10 9 8 7 6 5 4 3 2 1 0
Y:$FFFFAF
EMUX3 EMUX2 EMUX1 EMUX0
23 22 21 20 19 18 17 16 15 14 13 12
Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 9-2 EMUXR Register

Hardware and software reset clear all the bits of the EMUXR register. The selection of ESAI/ESAI_1 pins is shown in Table 9-1.

Table 9-1 EMUXR ESA/ESAI_1 Pin Selection

EMUXR bit		ESAI pin	ESAI_1 pin
EMUX0	0	SDO0 [PC11]	disconnected
EMUX0	1	disconnected	SDO0_1 [PE11]
EMUX1	0	SDO1[PC10]	disconnected
EMUX1	1	disconnected	SDO1_1 [PE10]
EMUX2	0	SDO2/SDI3 [PC9]	disconnected
EMUX2	1	disconnected	SDO2_1/SDI3_1 [PE9]
EMUX3	0	SDO3/SDI2 [PC8]	disconnected
EMUX3	1	disconnected	SDO3_1/SDI2_1 [PE8]

9.3.2 ESAI_1 Transmitter Clock Control Register (TCCR_1)

The read/write Transmitter Clock Control Register (TCCR_1) controls the ESAI_1 transmitter clock generator bit and frame sync rates, the bit rate and high frequency clock sources and the directions of the FST_1 and SCKT_1 signals. In synchronous mode, the bit clock defined for the transmitter determines the receiver bit clock as well. TCCR_1 also controls the number of words per frame for the serial data.

Hardware and software reset clear all the bits of the TCCR_1 register.

	11	10	9	8	7	6	5	4	3	2	1	0
Y:$FFFF96	TDC2	TDC1	TDC0	TPSR	TPM7	TPM6	TPM5	TPM4	TPM3	TPM2	TPM1	TPM0
	23	22	21	20	19	18	17	16	15	14	13	12
	THCKD	TFSD	TCKD	THCKP	TFSP	TCKP	TFP3	TFP2	TFP1	TFP0	TDC4	TDC3

Figure 9-3 TCCR_1 Register

9.3.2.1 TCCR_1 Tx High Freq. Clock Divider (TFP3-TFP0) - Bits 14–17

Since the ESAI_1 does not have the transmitter high frequency clock pin, the TFP3–TFP0 bits simply specify an additional division ratio in the clock divider chain. See Figure 9-4.

9.3.2.2 TCCR_1 Tx High Freq. Clock Polarity (THCKP) - Bit 20

The ESAI_1 does not have the transmitter high frequency clock pin. It it recommended that THCKP should be kept cleared.

9.3.2.3 TCCR_1 Tx High Freq. Clock Direction (THCKD) - Bit 23

The ESAI_1 does not have the transmitter high frequency clock pin. THCKD must be set for proper ESAI_1 transmitter section operation.

Table 9-2 Transmitter Clock Sources

THCKD	TFSD	TCKD	Transmitter Bit Clock Source	OUTPUTS	
0	X	X	Reserved		
1	0	0	SCKT_1		
1	0	1	INT		SCKT_1
1	1	0	SCKT_1	FST_1	
1	1	1	INT	FST_1	SCKT_1

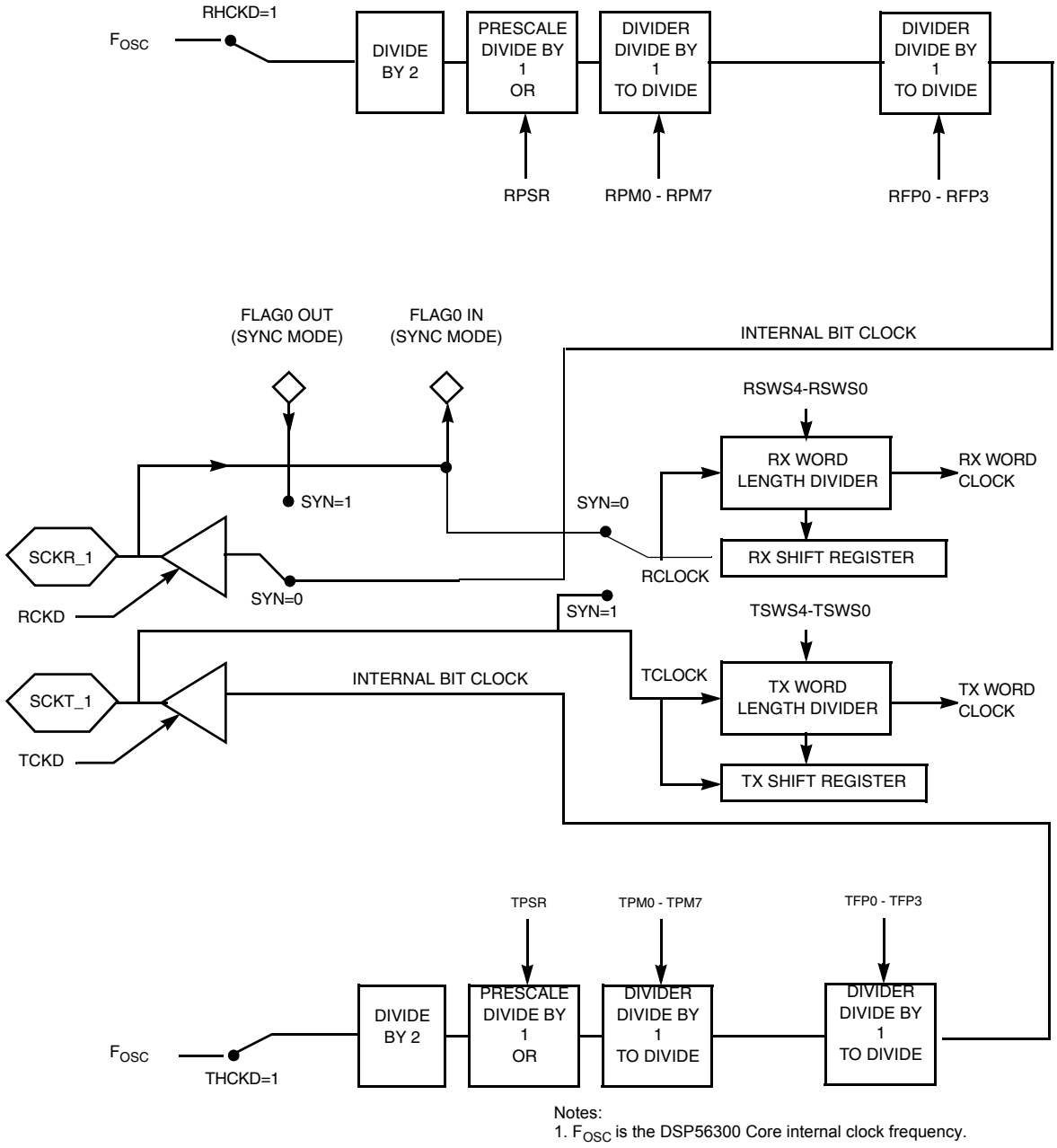


Figure 9-4 ESAI_1 Clock Generator Functional Block Diagram

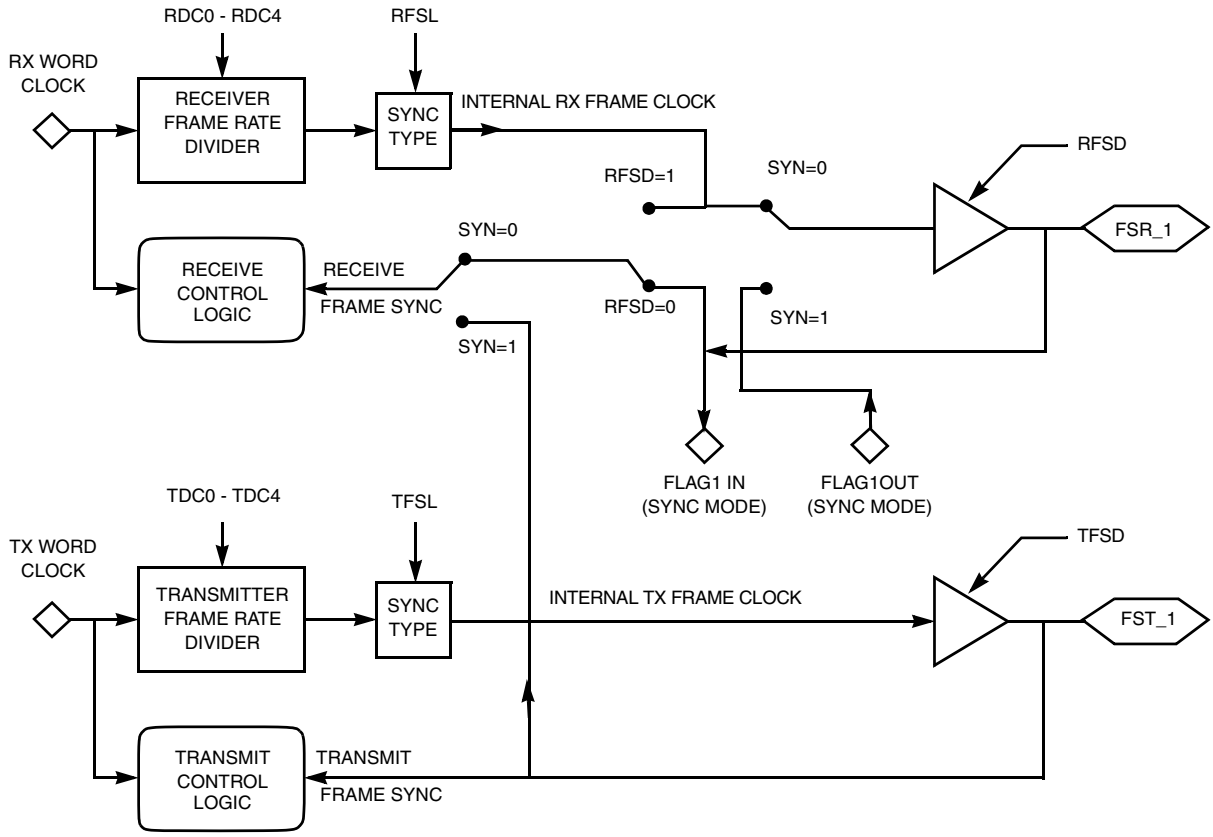


Figure 9-5 ESAI_1 Frame Sync Generator Functional Block Diagram

9.3.3 ESAI_1 Transmit Control Register (TCR_1)

The read/write Transmit Control Register (TCR_1) controls the ESAI_1 transmitter section. Interrupt enable bits for the transmitter section are provided in this control register. Operating modes are also selected in this register.

	11	10	9	8	7	6	5	4	3	2	1	0
Y:$FFFF95	TSWS1	TSWS0	TMOD1	TMOD0	TWA	TSHFD	TE5	TE4	TE3	TE2	TE1	TE0

	23	22	21	20	19	18	17	16	15	14	13	12
	TLIE	TIE	TEDIE	TEIE	TPR		PADC	TFSR	TFSL	TSWS4	TSWS3	TSWS2

Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 9-6 TCR_1 Register

Hardware and software reset clear all the bits in the TCR_1 register.

9.3.4 ESAI_1 Receive Clock Control Register (RCCR_1)

The read/write Receive Clock Control Register (RCCR_1) controls the ESAI_1 receiver clock generator bit and frame sync rates, word length, and number of words per frame for the serial data.

	11	10	9	8	7	6	5	4	3	2	1	0
Y:$FFFF98	RDC2	RDC1	RDC0	RPSR	RPM7	RPM6	RPM5	RPM4	RPM3	RPM2	RPM1	RPM0
	23	22	21	20	19	18	17	16	15	14	13	12
	RHCKD	RFSD	RCKD	RHCKP	RFSP	RCKP	RFP3	RFP2	RFP1	RFP0	RDC4	RDC3

Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 9-7 RCCR_1 Register

Hardware and software reset clear all the bits of the RCCR_1 register.

9.3.4.1 RCCR_1 Rx High Freq. Clock Divider (RFP3-RFP0) - Bits 14–17

Since the ESAI_1 does not have the receiver high frequency clock pin, the RFP3–RFP0 bits simply specify an additional division ratio in the clock divider chain. See Figure 9-4.

9.3.4.2 RCCR_1 Rx High Freq. Clock Polarity (RHCKP) - Bit 20

The ESAI_1 does not have the receiver high frequency clock pin. It it recommended that RHCKP should be kept cleared.

9.3.4.3 RCCR_1 Rx High Freq. Clock Direction (RHCKD) - Bit 23

The ESAI_1 does not have the receiver high frequency clock pin. RHCKD must be set for proper ESAI_1 receiver section operation.

Table 9-3 Receiver Clock Sources (asynchronous mode only)

RHCKD	RFSD	RCKD	Receiver Bit Clock Source	OUTPUTS	
0	X	X	Reserved		
1	0	0	SCKR_1		
1	0	1	INT		SCKR_1
1	1	0	SCKR_1	FSR_1	
1	1	1	INT	FSR_1	SCKR_1

9.3.5 ESAI_1 Receive Control Register (RCR_1)

The read/write Receive Control Register (RCR_1) controls the ESAI_1 receiver section.

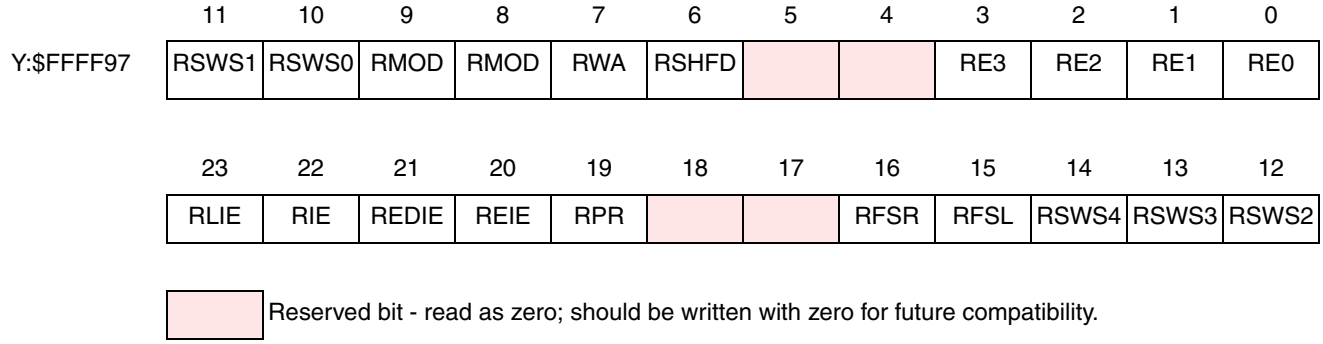

11 10 9 8 7 6 5 4 3 2 1 0
Y:$FFFF97
RSWS1 RSWS0 RMOD RMOD RWA RSHFD RE3 RE2 RE1 RE0
23 22 21 20 19 18 17 16 15 14 13 12
RLIE RIE REDIE REIE RPR RFSR RFSL RSWS4 RSWS3 RSWS2
Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 9-8 RCR_1 Register

Hardware and software reset clear all the bits in the RCR_1 register.

9.3.6 ESAI_1 Common Control Register (SAICR_1)

The read/write Common Control Register (SAICR_1) contains control bits for functions that use both the receive and transmit sections of the ESAI_1.

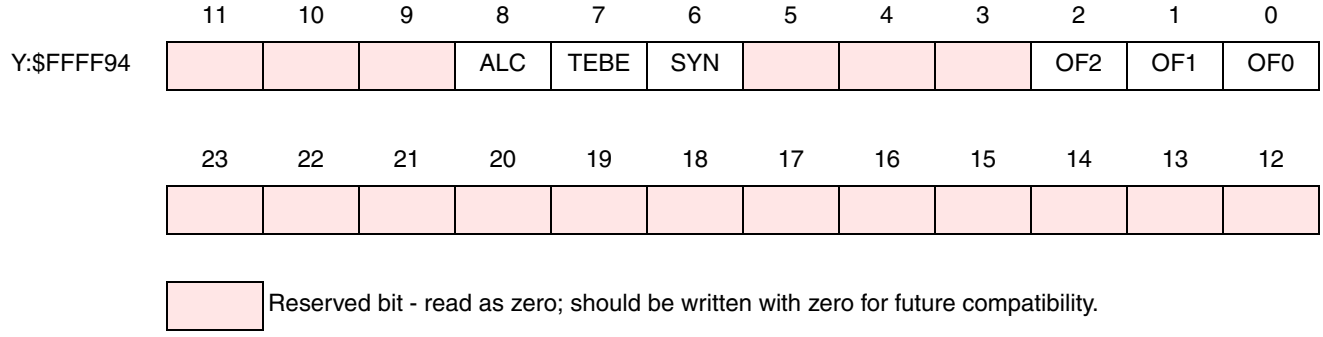

11 10 9 8 7 6 5 4 3 2 1 0
Y:$FFFF94
ALC TEBE SYN OF2 OF1 OF0
23 22 21 20 19 18 17 16 15 14 13 12
Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 9-9 SAICR_1 Register

Hardware and software reset clear all the bits in the SAICR_1 register.

9.3.7 ESAI_1 Status Register (SAISR_1)

The Status Register (SAISR_1) is a read-only status register used by the DSP to read the status and serial input flags of the ESAI_1.

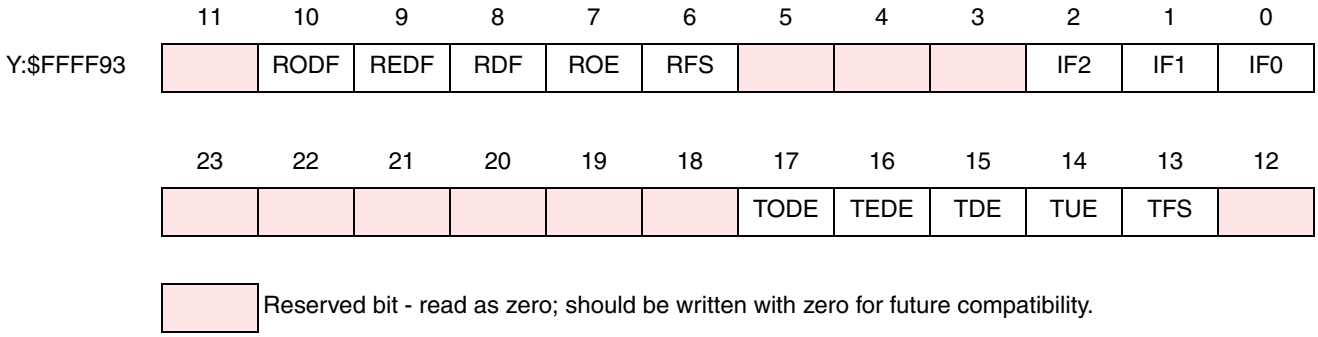

11 10 9 8 7 6 5 4 3 2 1 0
Y:$FFFF93
RODF REDF RDF ROE RFS IF2 IF1 IF0
23 22 21 20 19 18 17 16 15 14 13 12
TODE TEDE TDE TUE TFS
Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 9-10 SAISR_1 Register

9.3.8 ESAI_1 Receive Shift Registers

The receive shift registers receive the incoming data from the serial receive data pins. Data is shifted in by the selected (internal/external) bit clock when the associated frame sync I/O is asserted. Data is assumed to be received MSB first if RSHFD=0 and LSB first if RSHFD=1. Data is transferred to the ESAI_1 receive data registers after 8, 12, 16, 20, 24, or 32 serial clock cycles were counted, depending on the slot length control bits in the RCR_1 register.

9.3.9 ESAI_1 Receive Data Registers

The Receive Data Registers RX3_1, RX2_1, RX1_1, and RX0_1 are 24-bit read-only registers that accept data from the receive shift registers when they become full. The data occupies the most significant portion of the receive data registers, according to the ALC control bit setting. The unused bits (least significant portion, and 8 most significant bits when ALC=1) read as zeros. The DSP is interrupted whenever RXx_1 becomes full if the associated interrupt is enabled.

9.3.10 ESAI_1 Transmit Shift Registers

The Transmit Shift Registers contain the data being transmitted. Data is shifted out to the serial transmit data pins by the selected (internal/external) bit clock when the associated frame sync I/O is asserted. The number of bits shifted out before the shift registers are considered empty and may be written to again can be 8, 12, 16, 20, 24 or 32 bits (determined by the slot length control bits in the TCR_1 register). Data is shifted out of these registers MSB first if TSHFD=0 and LSB first if TSHFD=1.

9.3.11 ESAI_1 Transmit Data Registers

The Transmit Data registers TX5_1, TX4_1, TX3_1, TX2_1, TX1_1, and TX0_1 are 24-bit write-only registers. Data to be transmitted is written into these registers and is automatically transferred to the transmit shift registers. The data written (8, 12, 16, 20 or 24 bits) should occupy the most significant portion of the TXx_1 according to the ALC control bit setting. The unused bits (least significant portion, and the 8 most significant bits when ALC=1) of the TXx_1 are don’t care bits. The DSP is interrupted whenever the TXx_1 becomes empty if the transmit data register empty interrupt has been enabled.

9.3.12 ESAI_1 Time Slot Register (TSR_1)

The write-only Time Slot Register (TSR_1) is effectively a null data register that is used when the data is not to be transmitted in the available transmit time slot. The transmit data pins of all the enabled transmitters are in the high-impedance state for the respective time slot where TSR_1 has been written. The Transmitter External Buffer Enable pin (FSR_1 pin when SYN=1, TEBE=1, RFSD=1) disables the external buffers during the slot when the TSR_1 register has been written.

9.3.13 Transmit Slot Mask Registers (TSMA_1, TSMB_1)

The Transmit Slot Mask Registers (TSMA_1 and TSMB_1) are two read/write registers used by the transmitters in network mode to determine for each slot whether to transmit a data word and generate a transmitter empty condition (TDE=1), or to tri-state the transmitter data pins. TSMA_1 and TSMB_1 should each be considered as containing half a 32-bit register TSM_1. See Figure 9-11 and Figure 9-12. Bit number N in TSM_1 (TS**) is the enable/disable control bit for transmission in slot number N.

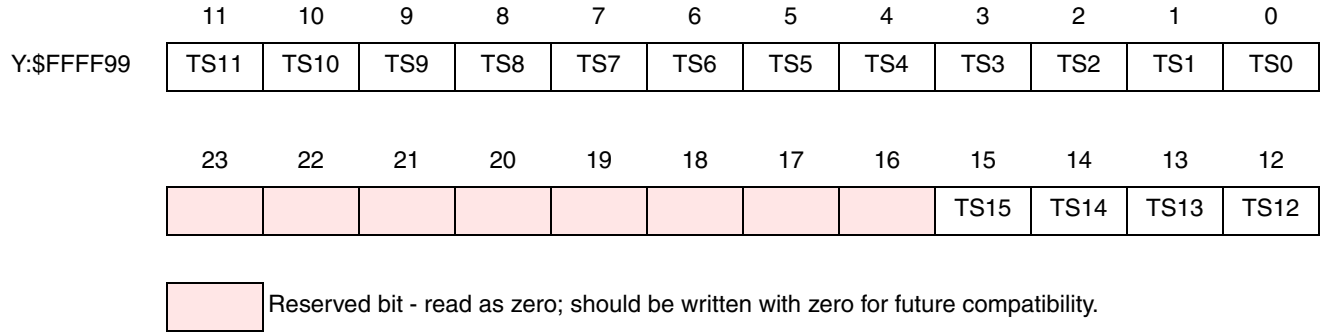

11 10 9 8 7 6 5 4 3 2 1 0
Y:$FFFF99
TS11 TS10 TS9 TS8 TS7 TS6 TS5 TS4 TS3 TS2 TS1 TS0
23 22 21 20 19 18 17 16 15 14 13 12
TS15 TS14 TS13 TS12
Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 9-11 TSMA_1 Register

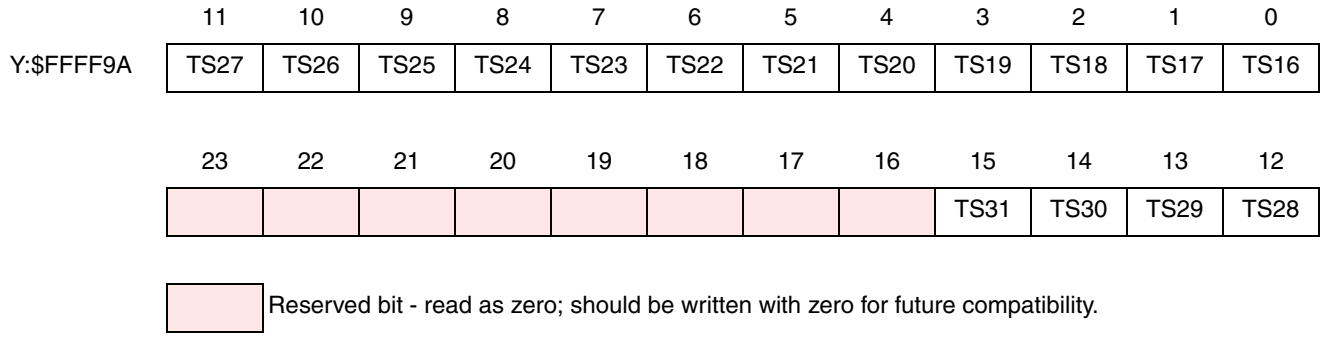

11 10 9 8 7 6 5 4 3 2 1 0
Y:$FFFF9A
TS27 TS26 TS25 TS24 TS23 TS22 TS21 TS20 TS19 TS18 TS17 TS16
23 22 21 20 19 18 17 16 15 14 13 12
TS31 TS30 TS29 TS28
Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 9-12 TSMB_1 Register

9.3.14 Receive Slot Mask Registers (RSMA_1, RSMB_1)

The Receive Slot Mask Registers (RSMA_1 and RSMB_1) are two read/write registers used by the receiver in network mode to determine for each slot whether to receive a data word and generate a receiver full condition (RDF=1), or to ignore the received data. RSMA_1 and RSMB_1 should be considered as each containing half of a 32-bit register RSM_1. See Figure 9-13 and Figure 9-14. Bit number N in RSM_1 (RS**) is an enable/disable control bit for receiving data in slot number N.

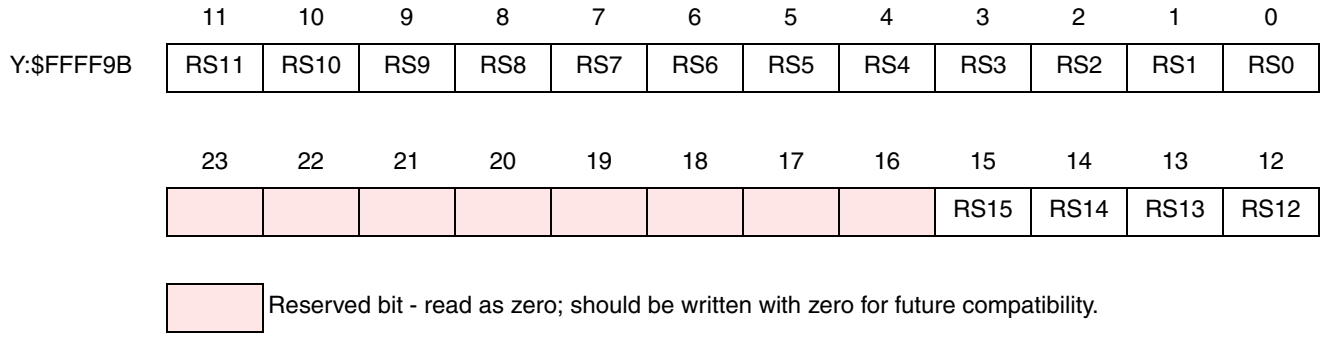

11 10 9 8 7 6 5 4 3 2 1 0
Y:$FFFF9B
RS11 RS10 RS9 RS8 RS7 RS6 RS5 RS4 RS3 RS2 RS1 RS0
23 22 21 20 19 18 17 16 15 14 13 12
RS15 RS14 RS13 RS12
Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 9-13 RSMA_1 Register

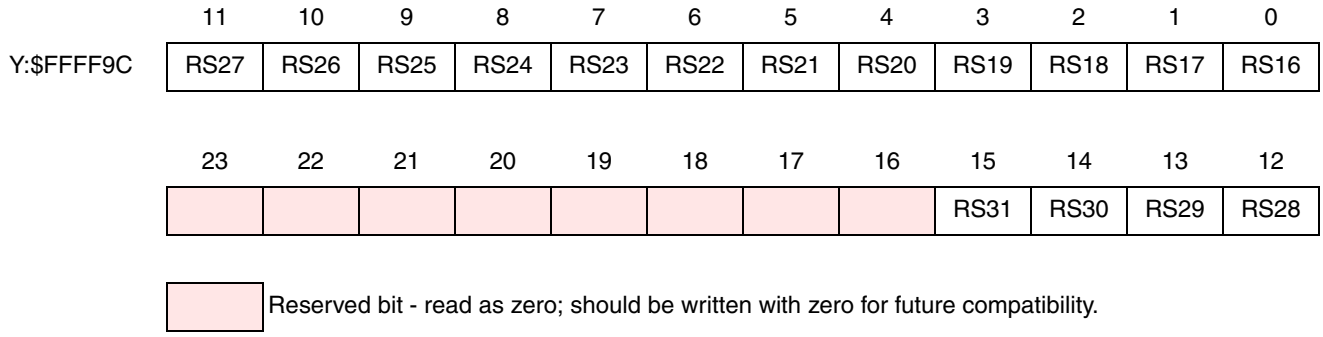

11 10 9 8 7 6 5 4 3 2 1 0
Y:$FFFF9C
RS27 RS26 RS25 RS24 RS23 RS22 RS21 RS20 RS19 RS18 RS17 RS16
23 22 21 20 19 18 17 16 15 14 13 12
RS31 RS30 RS29 RS28
Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 9-14 RSMB_1 Register

9.4 Operating Modes

9.4.1 ESAI_1 After Reset

Hardware or software reset clears the EMUXR register, the port E control register bits and the port E direction control register bits, which configure all 6 ESAI_1 dedicated I/O pins as disconnected, and all 4 shared pins as belonging to the ESAI. The ESAI_1 is in the individual reset state while all ESAI_1 signals are programmed as general-purpose I/O or disconnected, and is active only if at least one of the ESAI_1 I/O pins is programmed as belonging to the ESAI_1.

9.5 GPIO - Pins and Registers

The GPIO functionality of the ESAI_1 port is controlled by three registers: Port E Control register (PCRE), Port E Direction register (PRRE) and Port E Data register (PDRE).

9.5.1 Port E Control Register (PCRE)

The read/write 24-bit Port E Control Register (PCRE) in conjunction with the Port E Direction Register (PRRE) controls the functionality of the ESAI_1 GPIO pins. Each of the PE(11:0) bits controls the functionality of the corresponding port pin. See Table 9-4 for the port pin configurations. Hardware and software reset clear all PCRE bits.

9.5.2 Port E Direction Register (PRRE)

The read/write 24-bit Port E Direction Register (PRRE) in conjunction with the Port E Control Register (PCRE) controls the functionality of the ESAI_1 GPIO pins. Table 9-4 describes the port pin configurations. Hardware and software reset clear all PRRE bits.

Table 9-4 PCRE and PRRE Bits Functionality

PDE[i]	PE[i]	Port Pin[i] Function
0	0	disconnected
0	1	GPIO input
1	0	GPIO output
1	1	ESAI_1

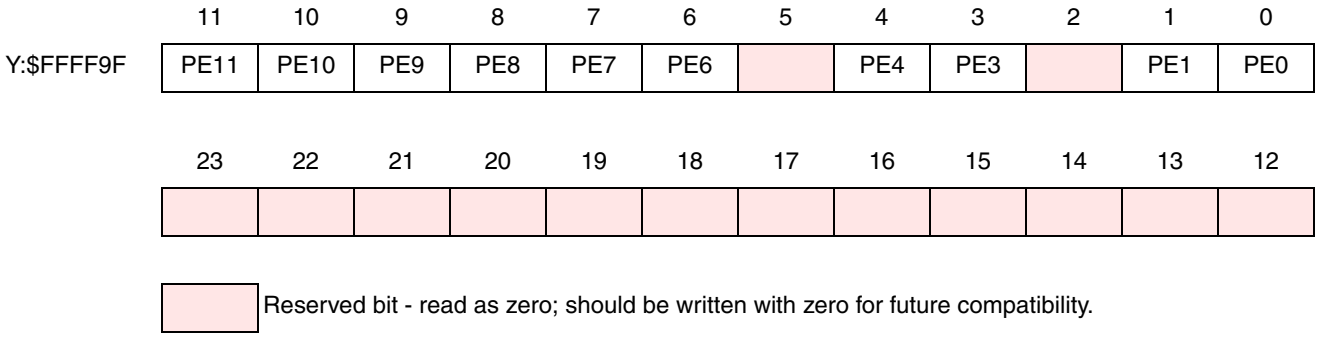

11 10 9 8 7 6 5 4 3 2 1 0
Y:$FFFF9F
PE11 PE10 PE9 PE8 PE7 PE6 PE4 PE3 PE1 PE0
23 22 21 20 19 18 17 16 15 14 13 12
Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 9-15 PCRE Register

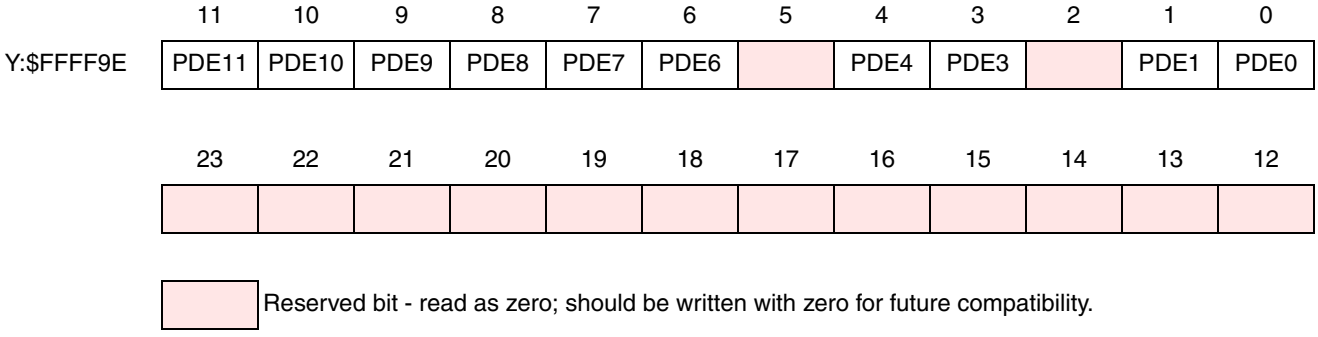

11 10 9 8 7 6 5 4 3 2 1 0
Y:$FFFF9E
PDE11 PDE10 PDE9 PDE8 PDE7 PDE6 PDE4 PDE3 PDE1 PDE0
23 22 21 20 19 18 17 16 15 14 13 12
Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 9-16 PRRE Register

9.5.3 Port E Data register (PDRE)

The read/write 24-bit Port E Data Register (see Figure 9-17) is used to read or write data to/from ESAI_1 GPIO pins. Bits PD(11:0) are used to read or write data from/to the corresponding port pins if they are configured as GPIO. If a port pin [i] is configured as a GPIO input, then the corresponding PD[i] bit will reflect the value present on this pin. If a port pin [i] is configured as a GPIO output, then the value written into the corresponding PD[i] bit will be reflected on this pin. If a port pin [i] is configured as disconnected, the corresponding PD[i] bit is not reset and contains undefined data.

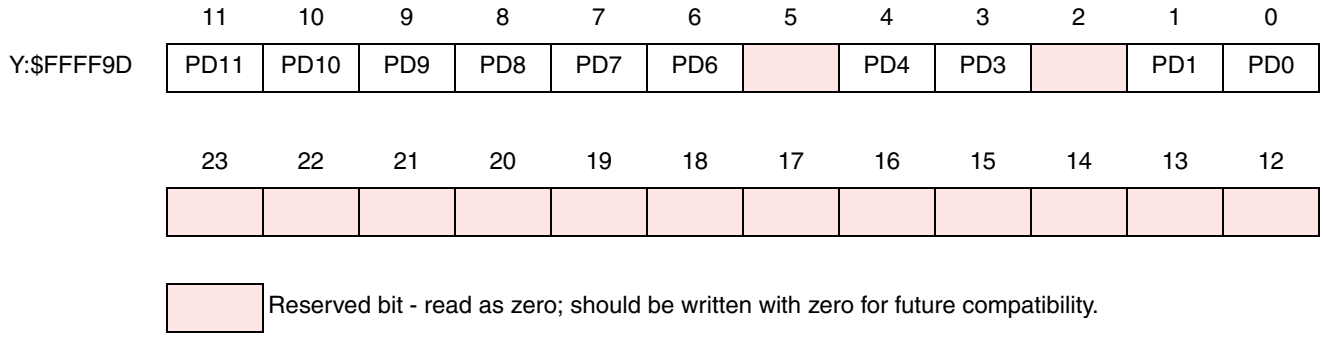

11 10 9 8 7 6 5 4 3 2 1 0
Y:$FFFF9D
PD11 PD10 PD9 PD8 PD7 PD6 PD4 PD3 PD1 PD0
23 22 21 20 19 18 17 16 15 14 13 12
Reserved bit - read as zero; should be written with zero for future compatibility.

Figure 9-17 PDRE Register

NOTES

10 Digital Audio Transmitter

10.1 Introduction

The Digital Audio Transmitter (DAX) is a serial audio interface module that outputs digital audio data in the AES/EBU, CP-340 and IEC958 formats. Some of the key features of the DAX are listed below.

- **Operates on a frame basis**—The DAX can handle one frame (consisting of two subframes) of audio and non-audio data at a time.
- **Double-buffered audio and non-audio data**—The DAX data path is double-buffered so the next frame data can be stored in the DAX without affecting the frame currently being transmitted.
- **Direct Memory Access**—Audio data and non-audio data can be written to the DAX using DMA.
- **Programmable clock source**—Users can select the DAX clock source, and this selection configures the DAX to operate in slave or master mode.
- **Supports both master mode and slave mode in a digital audio network**—If the user selects a divided DSP core clock, the DAX will operate in the master mode. If the user selects an external clock source, the DAX will operate in the slave mode.
- **GPIO**—Each of the two DAX pins can be configured as either GPIO or as specific DAX pin. Each pin is independent of the other. However, at least one of the two pins must be selected as a DAX pin to release the DAX from reset.

The accessible DAX registers are all mapped in the X I/O memory space. This allows programmers to access the DAX using standard instructions and addressing modes. Interrupts generated by the DAX can be handled with a fast interrupt for cases in which the non-audio data does not change from frame to frame. When the DAX interrupts are disabled, they can still be served by DMA or by a “polling” technique. A block diagram of the DAX is shown in Figure 10-1.

NOTE

The shaded registers in Figure 10-1 are directly accessible by DSP instructions.

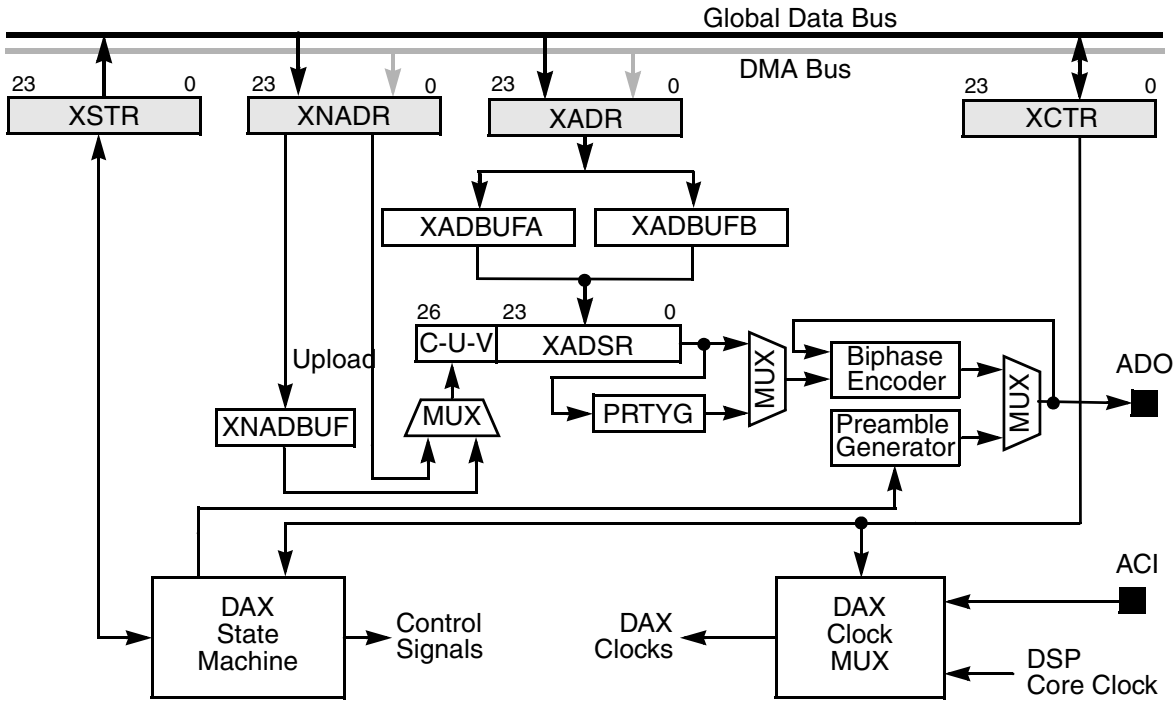


Figure 10-1 Digital Audio Transmitter (DAX) Block Diagram

10.2 DAX Signals

The DAX has two signal lines:

- **DAX Digital Audio Output (ADO/PD1)**—The ADO pin sends audio and non-audio data in the AES/EBU, CP340, and IEC958 formats in a biphase mark format. The ADO pin may also be used as a GPIO pin PD1 if the DAX is not operational.
- **DAX Clock Input (ACI/PD0)**—When the DAX clock is configured to be supplied externally, the external clock is applied to the ACI pin. The frequency of the external clock must be 256 times, 384 times, or 512 times the audio sampling frequency (256 × Fs, 384 × Fs, or 512 × Fs). The ACI pin may also be used as a GPIO pin PD0 when the DAX is disabled or when operating from the internal DSP clock.

10.3 DAX Functional Overview

The DAX consists of the following:

- Audio data register (XADR)
- Two audio data buffers (XADBUFA and XADBUFB)
- Non-audio data register (XNADR)
- Non-audio data buffer (XNADBUF)
- Audio and non-audio data shift register (XADSR)
- Control register (XCTR)
- Status register (XSTR)

- Parity generator (PRTYG)
- Preamble generator
- Biphase encoder
- Clock multiplexer
- Control state machine

XADR, XADBUFA, XADBUFB and XADSR creates a FIFO-like data path. Channel A is written to XADR and moves to XADBUFA. Then channel B is written to XADR, and when XADBUFB empties XADR moves into it. XADBUFA moves to the shift register XADSR when XADSR has shifted out its last bit. After channel A audio and non-audio data has been shifted out, XADBUFB moves into XADSR, and channel B audio and non audio shift begins.

The frame non-audio data (stored in XNADR) is transferred to the XADSR (for channel A) and to the XNADBUF registers (for channel B) at the beginning of a frame transmission. This is called an “upload.” The DAX audio data register empty (XADE) flag is set when XADR and XADBUFA are empty, and, if the audio data register empty interrupt is enabled (XDIE=1), an interrupt request is sent to the DSP core. The interrupt handling routine then sends the non-audio data bits to XNADR and the next frame of audio data to XADR (two subframes).

At the beginning of a frame transmission, one of the 8-bit channel A preambles (Z-preamble for the first subframe in a block, or X-preamble otherwise) is generated in the preamble generator, and then shifted out to the ADO pin in the first eight time slots. The preamble is generated in biphase mark format. The twenty-four audio and three non-audio data bits in the XADSR are shifted out to the biphase encoder, which shifts them out through the ADO pin in the biphase mark format in the next 54 time slots. The parity generator calculates an even parity over the 27 bits of audio and non-audio data, and then outputs the result through the biphase encoder to the ADO pin at the last two time slots. This is the end of the first (channel A) subframe transmission.

The second subframe transmission (channel B) starts with the preamble generator generating the channel B preamble (Y-preamble). At the same time, channel B audio and non-audio data is transferred to the XADSR shift-register from the XADBUFB and XNADBUF registers. The generated Y-preamble is output immediately after the channel A parity and is followed by the audio and non-audio data in the XADSR, which is in turn followed by the calculated parity for channel B. This completes a frame transmission. When the channel B parity is sent, the audio data for the next frame, stored in XADBUFA and the non-audio data bits from the XNADR, are uploaded to XADSR.

10.4 DAX Programming Model

The programmer-accessible DAX registers are shown in Figure 10-2. The registers are described in the following subsections. The Interrupt Vector table for the DAX is shown in Table 10-1. The internal interrupt priority is shown in Table 10-2.

Table 10-1 DAX Interrupt Vectors

Condition	Address	Description
XAUR	VBA:$28	DAX transmit underrun error
XADE & XBLK	VBA:$2A	DAX block transferred
XADE	VBA:$2E	DAX audio data register empty

Table 10-2 DAX Interrupt Priority

Priority	Interrupt
highest	DAX transmit underrun error
	DAX block transferred
lowest	DAX audio data register empty

10.5 DAX Internal Architecture

Hardware components shown in Figure 10-1 are described in the following sections. The DAX programming model is illustrated in Figure 10-2.

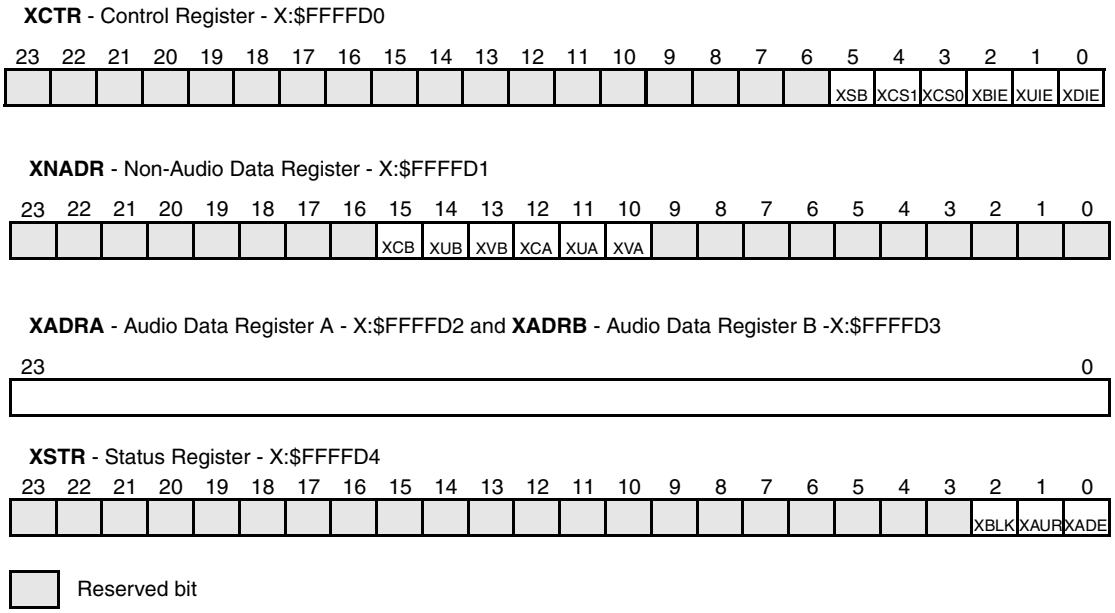

XCTR - Control Register - X:$FFFFD0
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
XSB XCS1 XCS0 XBIE XUIE XDIE
XNADR - Non-Audio Data Register - X:$FFFFD1
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
XCB XUB XVB XCA XUA XVA
XADRA - Audio Data Register A - X:$FFFFD2 and XADRB - Audio Data Register B -X:$FFFFD3
23 0
XSTR - Status Register - X:$FFFFD4
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
XBLK XAUR XADE
Reserved bit

Figure 10-2 DAX Programming Model

10.5.1 DAX Audio Data Register (XADR)

XADR is a 24-bit write-only register. One frame of audio data, which is to be transmitted in the next frame slot, is transferred to this register. Successive write accesses to this register will store channel A and channel B alternately in XADBUFA and in XADBUFB respectively. When XADR and XADBUFA are empty, XADE bit in the XSTR is set, and, if the audio data register empty interrupt is enabled (XDIE=1), an interrupt request is sent to the DSP core. When channel B is transferred to XADR, the XADE bit in the XSTR is cleared. XADR can also be accessed by DMA. When XADR and XADBUFA are empty, the DAX sends a DMA request to the core. The DMA first transfers non-audio data bits to XNADR (optional), then transfers channel A and channel B to XADR. The XADR can be accessed with two different successive addresses. This feature supports sending non-audio data bits, channel A and channel B to the DAX in three successive DMA transfers.

10.5.2 DAX Audio Data Buffers (XADBUFA / XADBUFB)

XADBUFA and XADBUFB are 24-bit registers that buffer XADR from XADSR, creating a FIFO-like data path. These registers hold the next two subframes of audio data to be transmitted. Channel A audio data is transferred from XADR to XADBUFA if XADBUFA is empty. Channel B audio data is transferred from XADR to XADBUFB if XADBUFB is empty. Audio data is transferred from XADBUFA and XADBUFB alternately to XADSR provided that XADSR shifted out all the audio and non-audio bits of the currently transmitted channel. This buffering mechanism provides more cycles for writing the next audio data to XADR. These registers are not directly accessible by DSP instructions.

10.5.3 DAX Audio Data Shift Register (XADSR)

The XADSR is a 27-bit shift register that shifts the 24-bit audio data and the 3-bit non-audio data for one subframe. The contents of XADBUFA or XADBUFB are directly transferred to the XADSR at the beginning of the subframe transmission. The channel A subframe is transferred to XADSR at the same time that the three bits of non-audio data (V-bit, U-bit and C-bit) for channel A in the DAX non-audio data register (XNADR) are transferred to the three highest-order bits of the XADSR. At the beginning of the channel B transmission, audio and non-audio data for channel B are transferred from the XADBUFB and the XNADBUF to the XADSR for shifting. The data in the XADSR is shifted toward the lowest-order bit at the fifth to thirty-first bit slot of each subframe transmission. This register is not directly accessible by DSP instructions.

10.5.4 DAX Non-Audio Data Register (XNADR)

The XNADR is a 24-bit write-only register. It holds the three bits of non-audio data for each subframe. XNADR can be accessed by core instructions or by DMA. The contents of the XNADR are shown in Figure 10-2. XNADR is not affected by any of the DAX reset states. The XNADR bits are described in the following paragraphs.

10.5.4.1 DAX Channel A Validity (XVA)—Bit 10

The value of the XVA bit is transmitted as the twenty-ninth bit (Bit 28) of channel A subframe in the next frame.

10.5.4.2 DAX Channel A User Data (XUA)—Bit 11

The value of the XUA bit is transmitted as the thirtieth bit (Bit 29) of the channel A subframe in the next frame.

10.5.4.3 DAX Channel A Channel Status (XCA)—Bit 12

The value of the XCA bit is transmitted as the thirty-first bit (Bit 30) of the channel A subframe in the next frame.

10.5.4.4 DAX Channel B Validity (XVB)—Bit 13

The value of the XVB bit is transmitted as the twenty-ninth bit (Bit 28) of the channel B subframe in the next frame.

10.5.4.5 DAX Channel B User Data (XUB)—Bit 14

The value of the XUB bit is transmitted as the thirtieth bit (Bit 29) of the channel B subframe in the next frame.

10.5.4.6 DAX Channel B Channel Status (XCB)—Bit 15

The value of the XCB bit is transmitted as the thirty-first bit (Bit 30) of the channel B subframe in the next frame.

10.5.4.7 XNADR Reserved Bits—Bits 0-9, 16–23

These XNADR bits are reserved. They read as 0, and should be written with 0 to ensure compatibility with future device versions.

10.5.5 DAX Non-Audio Data Buffer (XNADBUF)

The XNADBUF is a 3-bit register that temporarily holds channel B non-audio data (XVB, XUB and XCB) for the current transmission while the channel A data is being transmitted. This mechanism provides programmers more instruction cycles to store the next frame's non-audio data to the XCB, XUB, XVB, XCA, XUA and XVA bits in the XNADR. The data in the XNADBUF register is transferred to the XADSR along with the contents of the XADBUF register at the beginning of channel B transmission.

NOTE

The XNADBUF register is not directly accessible by DSP instructions.

10.5.6 DAX Control Register (XCTR)

The XCTR is a 24-bit read/write register that controls the DAX operation. The contents of the XCTR are shown in Figure 10-2. XCTR is cleared by software reset and hardware reset. The XCTR bits are described in the following paragraphs.

10.5.6.1 Audio Data Register Empty Interrupt Enable (XDIE)—Bit 0

When the XDIE bit is set, the audio data register empty interrupt is enabled and sends an interrupt request signal to the DSP if the XADE status bit is set. When XDIE bit is cleared, this interrupt is disabled.

10.5.6.2 Underrun Error Interrupt Enable (XUIE)—Bit 1

When the XUIE bit is set, the underrun error interrupt is enabled and sends an interrupt request signal to the DSP if the XAUR status bit is set. When XUIE bit is cleared, this interrupt is disabled.

10.5.6.3 Block Transferred Interrupt Enable (XBIE)—Bit 2

When the XBIE bit is set, the block transferred interrupt is enabled and sends an interrupt request signal to the DSP if the XBLK and XADE status bits are set. When XBIE bit is cleared, this interrupt is disabled.

10.5.6.4 DAX Clock Input Select (XCS[1:0])—Bits 3–4

The XCS[1:0] bits select the source of the DAX clock and/or its frequency. Table 10-3 shows the configurations selected by these bits. These bits should be changed only when the DAX is disabled.

Table 10-3 Clock Source Selection

XCS1	XCS0	DAX Clock Source
0	0	DSP Core Clock (f = 1024 **X** fs)
0	1	ACI Pin, f = 256 **X** fs
1	0	ACI Pin, f = 384 **X** fs
1	1	ACI Pin, f = 512 **X** fs

10.5.6.5 DAX Start Block (XSB)—Bit 5

The XSB bit forces the DAX to start a new block. When this bit is set, the next frame will start with “Z” preamble and will start a new block even though the current block was not finished. This bit is cleared when the new block starts.

10.5.6.6 XCTR Reserved Bits—Bits 6-23

These XCTR bits are reserved. They read as 0 and should be written with 0 for future compatibility.

10.5.7 DAX Status Register (XSTR)

The XSTR is a 24-bit read-only register that contains the DAX status flags. The contents of the XSTR are shown in Figure 10-2. XSTR is cleared by software reset, hardware reset an by the stop state. The XSTR bits are described in the following paragraphs.

10.5.7.1 DAX Audio Data Register Empty (XADE)—Bit 0

The XADE status flag indicates that the DAX audio data register XADR and the audio data buffer XADBUFA are empty (and ready to receive the next frame's audio data). This bit is set at the beginning of every frame transmission (more precisely, when channel A audio data is transferred from XADBUFA to XADSR). When XADE is set and the interrupt is enabled (XDIE = 1), an audio data register empty interrupt request is sent to the DSP core. XADE is cleared by writing two channels of audio data to XADR.

10.5.7.2 DAX Transmit Underrun Error Flag (XAUR)—Bit 1

The XAUR status flag is set when the DAX audio data buffers XADBUFA or XADBUFB are empty and the respective audio data upload occurs. When a DAX underrun error occurs, the previous frame data will be retransmitted in both channels. When XAUR is set and the interrupt is enabled (XUIE = 1), an underrun error interrupt request is sent to the DSP core. This allows programmers to write an exception handling routine for this special case. The XAUR bit is cleared by reading the XSTR register with XAUR set, followed by writing two channels of audio data to XADR.

10.5.7.3 DAX Block Transfer Flag (XBLK)—Bit 2

The XBLK flag indicates that the frame being transmitted is the last frame in a block. This bit is set at the beginning of the transmission of the last frame (the 191st frame). This bit does not cause any interrupt. However, if XBIE=1 it causes a change in the interrupt vector sent to DSP core in the event of an audio data register empty interrupt, so that a different interrupt routine can be called (providing the next non-audio data structures for the next block as well as storing audio data for the next frame). Writing two channels of audio data to XADR clears this bit.

The relative timing of transmit frames and XADE and XBLK flags is shown in Figure 10-3.

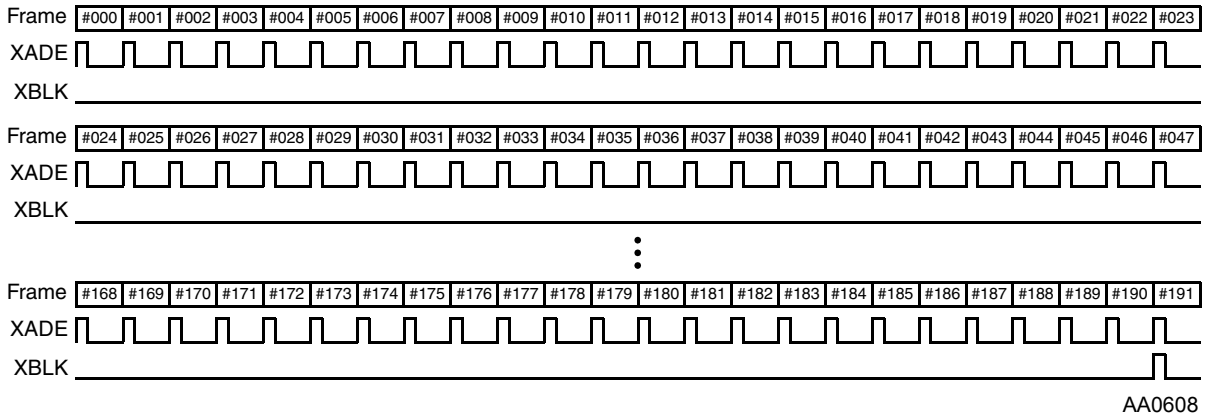


Figure 10-3 DAX Relative Timing

10.5.7.4 XSTR Reserved Bits—Bits 3–23

These XSTR bits are reserved. They read as 0, and should be written with 0 to ensure compatibility with future device versions.

10.5.8 DAX Parity Generator (PRTYG)

The PRTYG generates the parity bit for the subframe being transmitted. The generated parity bit ensures that subframe bits four to thirty-one will carry an even number of ones and zeroes.

10.5.9 DAX Biphase Encoder

The DAX biphase encoder encodes each audio and non-audio bit into its biphase mark format and shifts this encoded data out to the ADO output pin synchronously to the biphase clock.

10.5.10 DAX Preamble Generator

The DAX preamble generator automatically generates one of three preambles in the 8-bit preamble shift register at the beginning of each subframe transmission, and shifts it out. The generated preambles always start with “0”. Bit patterns of preambles generated in the preamble generator are shown in Table 10-4. The preamble bits are already in the biphase mark format.

Table 10-4 Preamble Bit Patterns

Preamble	Bit Pattern	Channel
X	00011101	A
Y	00011011	B
Z	00010111	A (first in block)

There is no programmable control for the preamble selection. The first subframe to be transmitted (immediately after the DAX is enabled) is the beginning of a block, and therefore it has a “Z” preamble. This is followed by the second subframe, which has an “Y” preamble. After that, “X” and “Y” preambles are transmitted alternately until the end of the block transfer (192 frames transmitted). See Figure 10-4 for an illustration of the preamble sequence.

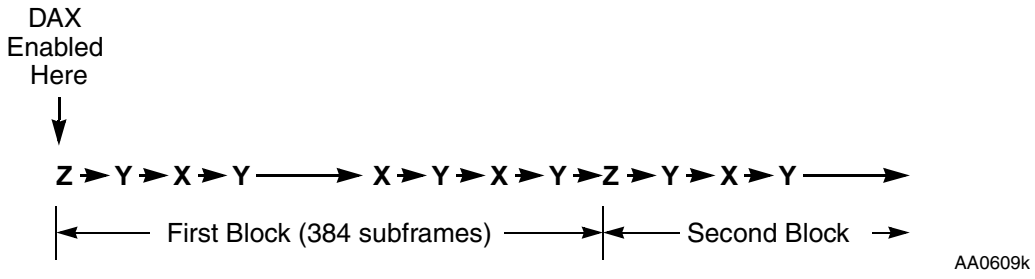


Figure 10-4 Preamble sequence

10.5.11 DAX Clock Multiplexer

The DAX clock multiplexer selects one of the clock sources and generates the biphase clock (128 × Fs) and shift clock (64 × Fs). The clock source can be selected from the following options (see also Section 10.5.6.4, "DAX Clock Input Select (XCS[1:0])—Bits 3–4").

- The internal DSP core clock—assumes 1024 × Fs
- DAX clock input pin (ACI)—512 × Fs
- DAX clock input pin (ACI)—384 × Fs
- DAX clock input pin (ACI)—256 × Fs

Figure 10-5 shows how each clock is divided to generate the biphase and bit shift clocks

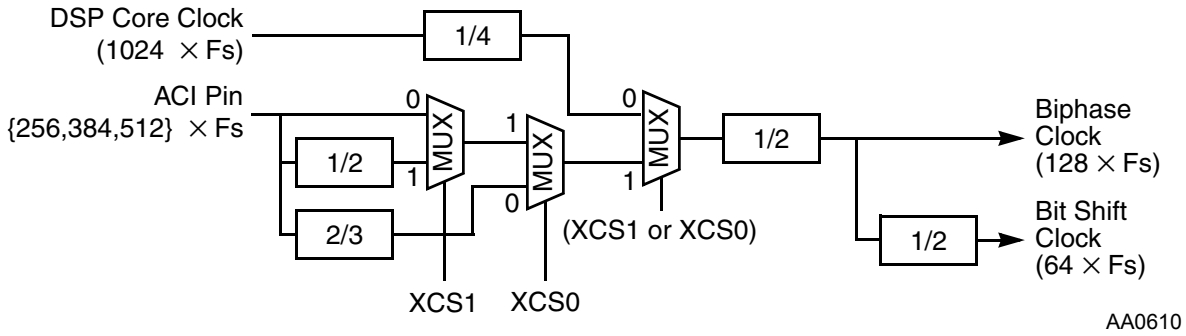


Figure 10-5 Clock Multiplexer Diagram

NOTE

For proper operation of the DAX, the DSP core clock frequency must be at least five times higher than the DAX bit shift clock frequency (64 × Fs).

10.5.12 DAX State Machine

The DAX state machine generates a set of sequencing signals used in the DAX.

10.6 DAX Programming Considerations

The following sections describe programming considerations for the DAX.

10.6.1 Initiating A Transmit Session

To initiate the DAX operation, follow this procedure:

1. Ensure that the DAX is disabled (PC1 and PC0 bits of port control register PCR are cleared)
2. Write the non-audio data to the corresponding bits in the XNADR register
3. Write the channel A and channel B audio data in the XADR register
4. Write the transmit mode to the XCTR register
5. Enable DAX by setting PC1 bit (and by setting PC0 bit if in slave mode) in the port control register (PCR); transmission begins.

10.6.2 Audio Data Register Empty Interrupt Handling

When the XDIE bit is set and the DAX is active, an audio data register empty interrupt (XADE = 1) is generated once at the beginning of every frame transmission. Typically, within an XADE interrupt, the

non-audio data bits of the next frame are stored in XNADR and one frame of audio data to be transmitted in the next frame is stored in the FIFO by two consecutive MOVEP instructions to XADR. If the non-audio bits are not changed from frame to frame, this procedure can be handled within a fast interrupt routine. Storing the next frame's audio data in the FIFO clears the XADE bit in the XSTR.

10.6.3 Block Transferred Interrupt Handling

An interrupt with the XBLK vector indicates the end of a block transmission and may require some computation to provide the next non-audio data structures that are to be transmitted within the next block. Within the routine, the next audio data can be stored in the FIFO by two consecutive MOVEP instructions to XADR, and the next non-audio data can be stored in the XNADR. The XBLK interrupt occurs only if the XBIE bit in XCTR is set. If XBIE is cleared, a XADE interrupt vector will take place.

10.6.4 DAX operation with DMA

During DMA transfers, the XDIE bit of the XCTR must be cleared to avoid XADE interrupt services by the DSP core. The initialization appearing in Section 10.6.1, "Initiating A Transmit Session" is relevant for DMA operation. DMA transfers can be performed with or without changing non-audio bits from frame to frame. Table 10-5 describes two examples of DMA configuration.

Table 10-5 Examples of DMA configuration

Register	Non-audio data bits change	Non-audio data bits do not change
DCR2	DE=1; Enable DMA channel. DIE=1; Enable DMA interrupt. DTM[2:0]=010; Line transfer mode. D3D=0; Not 3D. DAM[5:3]=000; 2D mode. DAM[2:0]=101; post increment by 1. DDS[1:0]=00; X memory space. DRS[4:0]=01010; DAX is DMA request source. Other bits are application dependent.	DE=1; Enable DMA channel. DIE=1; Enable DMA interrupt. DTM[2:0]=010; Line transfer mode. D3D=0; Not 3D. DAM[5:3]=000; 2D mode. DAM[2:0]=101; post increment by 1. DDS[1:0]=00; X memory space. DRS[4:0]=01010; DAX is DMA request source. Other bits are application dependent.
DCO2	DCOH=number of frames in block - 1 DCOL=$002; 3 destination registers	DCOH=number of frames in block - 1 DCOL=$001; 2 destination registers
DSR2	first memory address of the block	first memory address of the block
DDR2	XNADR address (base address + $1)	XADR address (base address + $2)
DOR0	$FFFFFE; offset=-2	$FFFFFF; offset=-1

The memory organization employed for DMA transfers depends on whether or not non-audio data changes from frame to frame as shown in Figure 10-6.

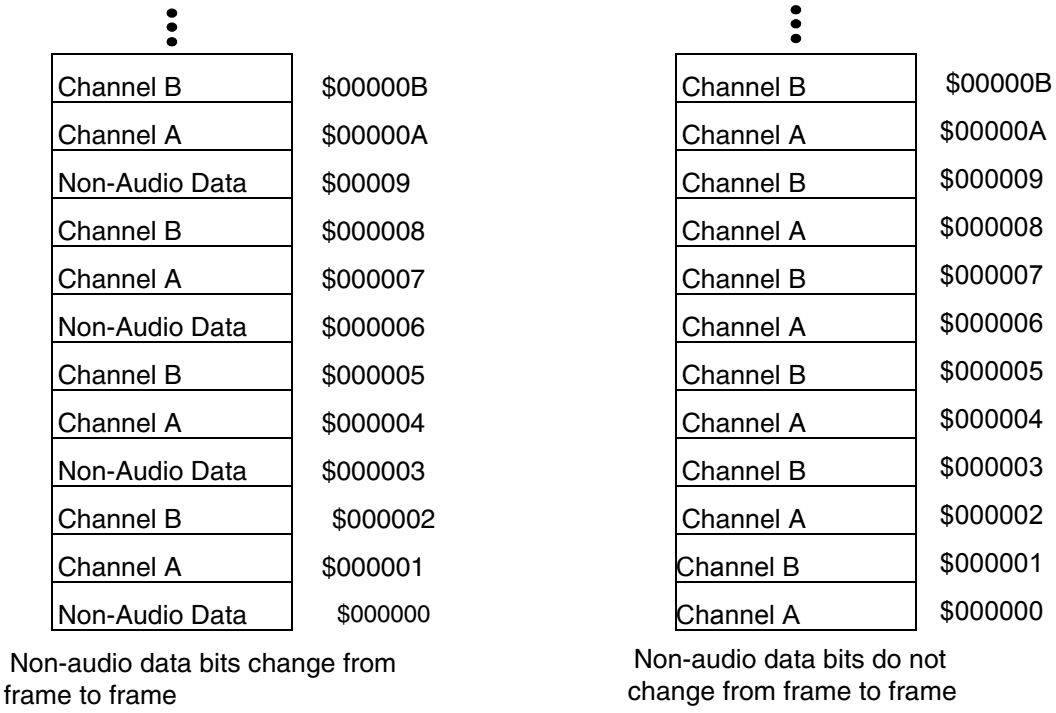


Figure 10-6 Examples of data organization in memory

10.6.5 DAX Operation During Stop

The DAX operation cannot continue when the DSP is in the stop state since no DSP clocks are active. While the DSP is in the stop state, the DAX will remain in the individual reset state and the status flags are initialized as described for resets. No DAX control bits are affected. The DAX should be disabled before the DSP enters the stop state.

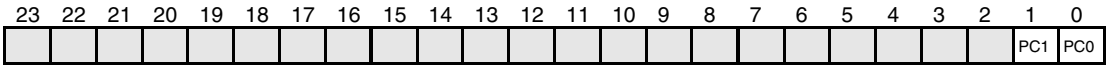
10.7 GPIO (PORT D) - Pins and Registers

The Port D GPIO functionality of the DAX is controlled by three registers: Port D Control Register (PCRD), Port D Direction Register (PRRD) and Port D Data Register (PDRD).

10.7.1 Port D Control Register (PCRD)

The read/write 24-bit DAX Port D Control Register controls the functionality of the DAX GPIO pins. Each of the PC[1:0] bits controls the functionality of the corresponding port pin. When a PC[i] bit is set, the corresponding port pin is configured as a DAX pin. When a PC[i] bit is cleared, the corresponding port pin is configured as GPIO pin. If both PC1 and PC0 are cleared, the DAX is disabled. Hardware and software reset clear all PCRD bits.

PCRD -Port D Control Register - X:$FFFFD7

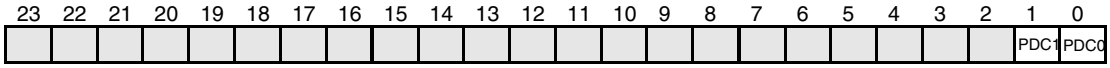


read as zero, should be written with zero for future compatibility

10.7.2 Port D Direction Register (PRRD)

The read/write 24-bit Port D Direction Register controls the direction of the DAX GPIO pins. When port pin[i] is configured as GPIO, PDC[i] controls the port pin direction. When PDC[i] is set, the GPIO port pin[i] is configured as output. When PDC[i] is cleared the GPIO port pin[i] is configured as input. Hardware and software reset clear all PRRD bits. Table 10-6 describes the port pin configurations.

PRRD - Port D Direction Register - X:$FFFFD6



read as zero, should be written with zero for future compatibility

Figure 10-7 Port D Direction Register (PRRD)

Table 10-6 DAX Port GPIO Control Register Functionality

PDC1	PC1	ADO/PD1 pin	PDC0	PC0	ACI/PD0 pin	DAX state
0	0	*Disconnected*	0	0	*Disconnected*	Personal Reset
0	0	*Disconnected*	0	1	PD0 Input	Personal Reset
0	0	*Disconnected*	1	0	PD0 Output	Personal Reset
0	0	*Disconnected*	1	1	ACI	Enabled
0	1	PD1 Input	0	0	*Disconnected*	Personal Reset
0	1	PD1 Input	0	1	PD0 Input	Personal Reset
0	1	PD1 Input	1	0	PD0 Output	Personal Reset
0	1	PD1 Input	1	1	ACI	Enabled
1	0	PD1 Output	0	0	*Disconnected*	Personal Reset
1	0	PD1 Output	0	1	PD0 Input	Personal Reset
1	0	PD1 Output	1	0	PD0 Output	Personal Reset
1	0	PD1 Output	1	1	ACI	Enabled
1	1	ADO	0	0	*Disconnected*	Enabled
1	1	ADO	0	1	PD0 Input	Enabled

Table 10-6 DAX Port GPIO Control Register Functionality

PDC1	PC1	ADO/PD1 pin	PDC0	PC0	ACI/PD0 pin	DAX state
1	1	ADO	1	0	PD0 Output	Enabled
1	1	ADO	1	1	ACI	Enabled

10.7.3 Port D Data Register (PDRD)

The read/write 24-bit Port D Data Register is used to read or write data to/from the DAX GPIO pins. Bits PD[1:0] are used to read or write data from/to the corresponding port pins if they are configured as GPIO. If a port pin [i] is configured as a GPIO input, then the corresponding PD[i] bit will reflect the value present on this pin. If a port pin [i] is configured as a GPIO output, then the value written into the corresponding PD[i] bit will be reflected on the this pin. Hardware and software reset clear all PDRD bits.

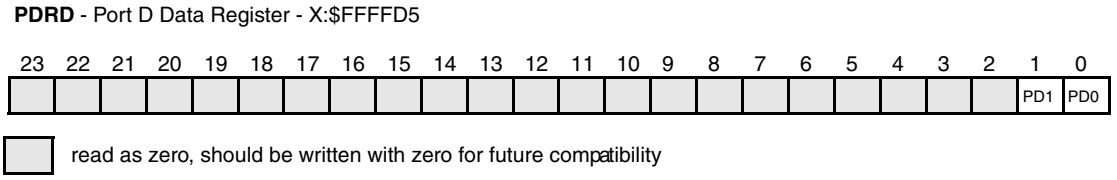

PDRD - Port D Data Register - X:$FFFFD5
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
PD1
PD0
read as zero, should be written with zero for future compatibility

Figure 10-8 Port D Data Register (PDRD)

11 Timer/ Event Counter

11.1 Introduction

This section describes the internal timer/event counter in the DSP56366. Each of the three timers (timer 0, 1 and 2) can use internal clocking to interrupt the DSP56366 or trigger DMA transfers after a specified number of events (clocks). In addition, timer 0 provides external access via the bidirectional signal TIO0.

When the TIO0 pin is configured as an input, timer 0 can count or capture events, or measure the width or period of an external signal. When TIO0 is configured as an output, timer 0 can function as a timer, a watchdog timer, or a pulse width modulator. TIO0 can also function as a GPIO signal.

11.2 Timer/Event Counter Architecture

The timer module is composed of a common 21-bit prescaler and three independent general purpose 24-bit timer/event counters, each having its own register set.

11.2.1 Timer/Event Counter Block Diagram

Figure 11-1 shows a block diagram of the timer/event counter. This module includes a 24-bit timer prescaler load register (TPLR), a 24-bit timer prescaler count register (TPCR), a 21-bit prescaler clock counter, and three timers. Each of the three timers may use the prescaler clock as its clock source.

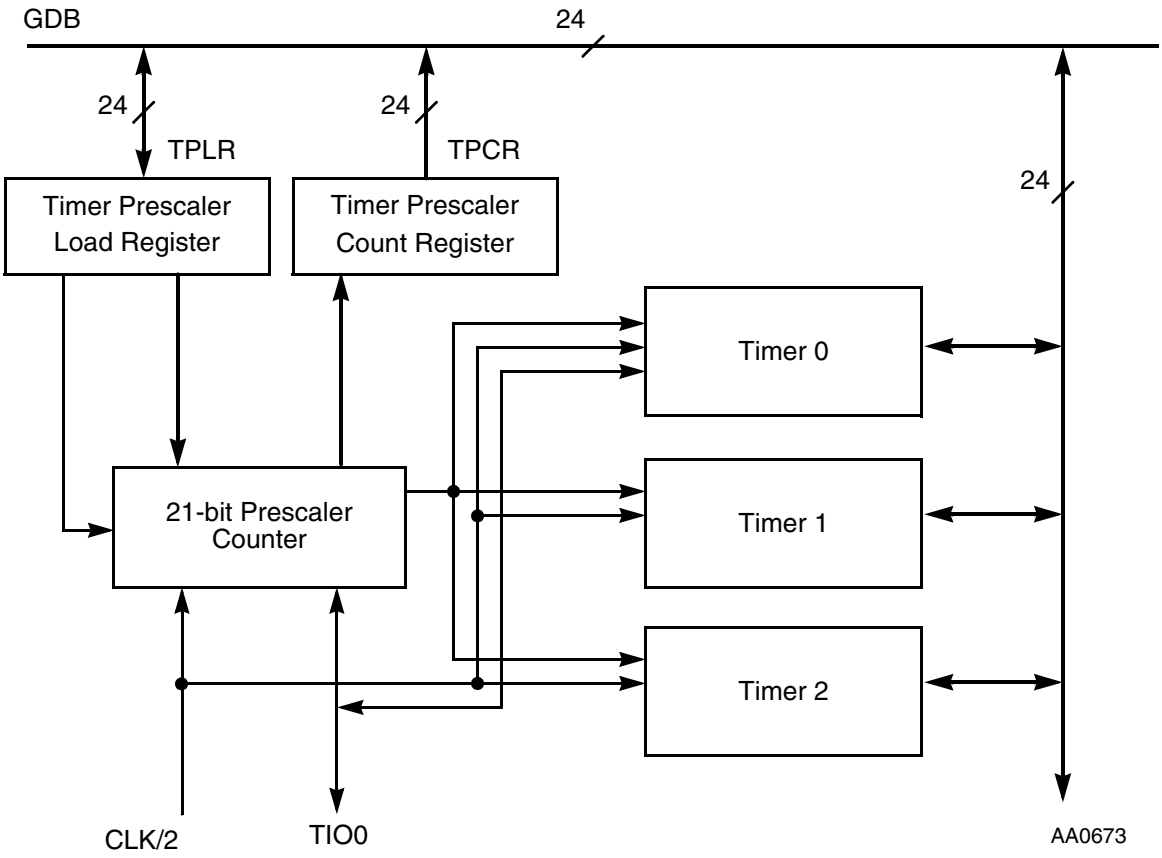


Figure 11-1 Timer/Event Counter Block Diagram

11.2.2 Individual Timer Block Diagram

Figure 11-2 shows the structure of an individual timer module. The three timers are identical in structure, but only timer 0 is externally accessible.

Each timer includes a 24-bit counter, a 24-bit read/write timer control and status register (TCSR), a 24-bit read-only timer count register (TCR), a 24-bit write-only timer load register (TLR), a 24-bit read/write timer compare register (TCPR), and logic for clock selection and interrupt/DMA trigger generation.

The timer mode is controlled by the TC[3:0] bits of the timer control/status register (TCSR). Timer modes are described in Section 11.4, "Timer Modes of Operation".

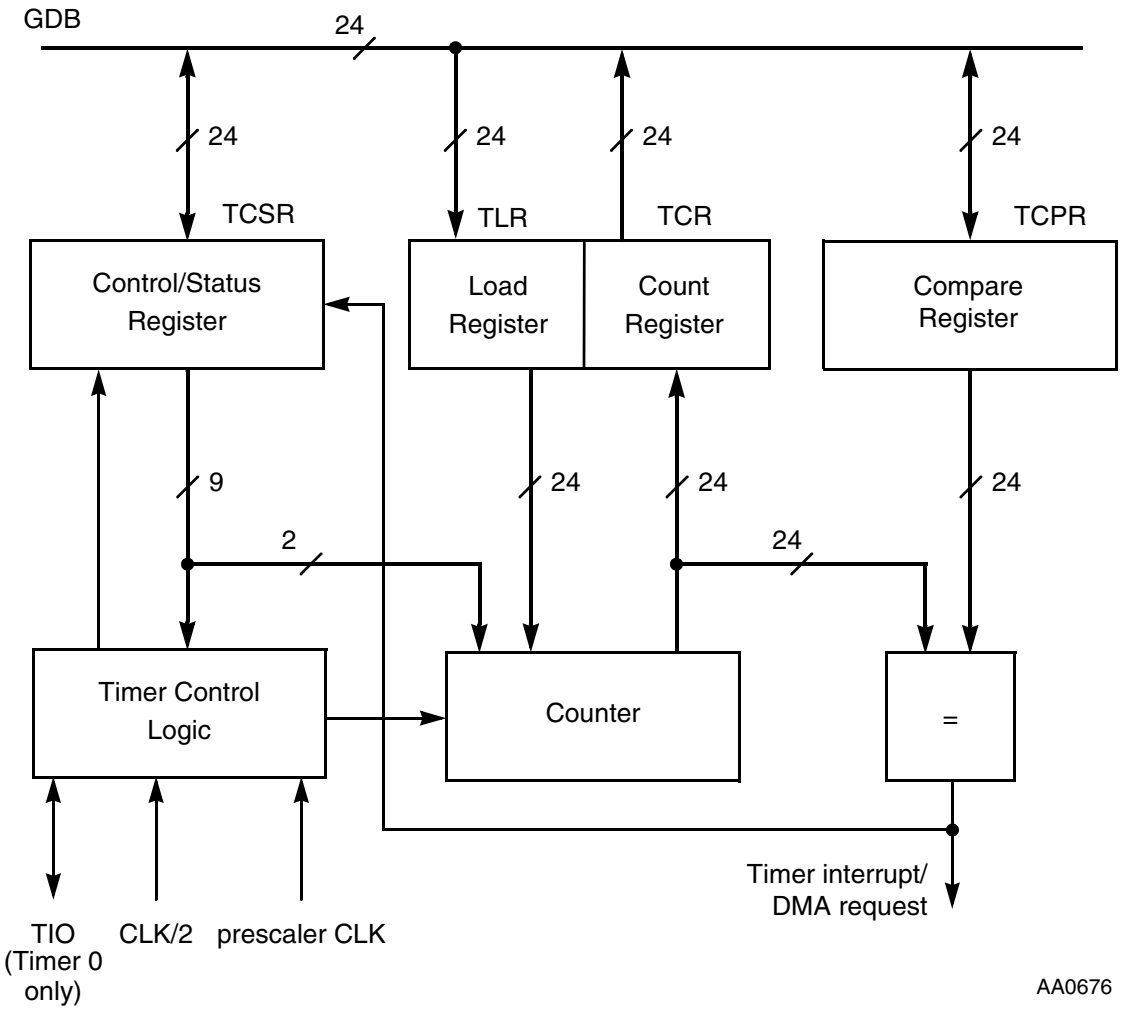


Figure 11-2 Timer Block Diagram

11.3 Timer/Event Counter Programming Model

The DSP56366 views each timer as a memory-mapped peripheral with four registers occupying four 24-bit words in the X data memory space. Either standard polled or interrupt programming techniques can be used to service the timers. The timer programming model is shown in Figure 11-3.

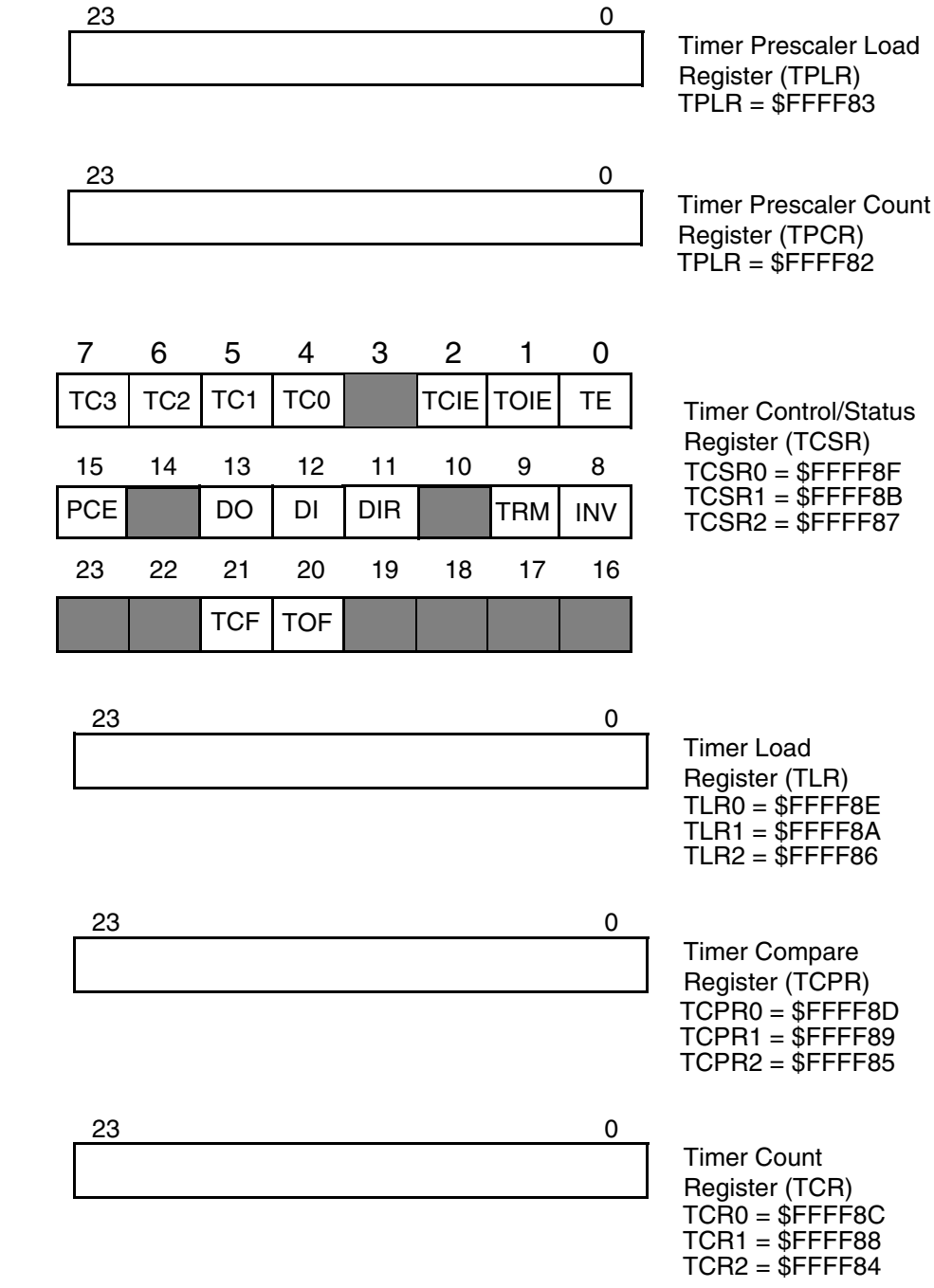


Figure 11-3 Timer Module Programmer's Model

11.3.1 Prescaler Counter

The prescaler counter is a 21-bit counter that is decremented on the rising edge of the prescaler input clock. The counter is enabled when at least one of the three timers is enabled (i.e., one or more of the timer enable (TE) bits are set) and is using the prescaler output as its source (i.e., one or more of the PCE bits are set).

11.3.2 Timer Prescaler Load Register (TPLR)

The TPLR is a 24-bit read/write register that controls the prescaler divide factor (i.e., the number that the prescaler counter will load and begin counting from) and the source for the prescaler input clock. See Figure 11-4.

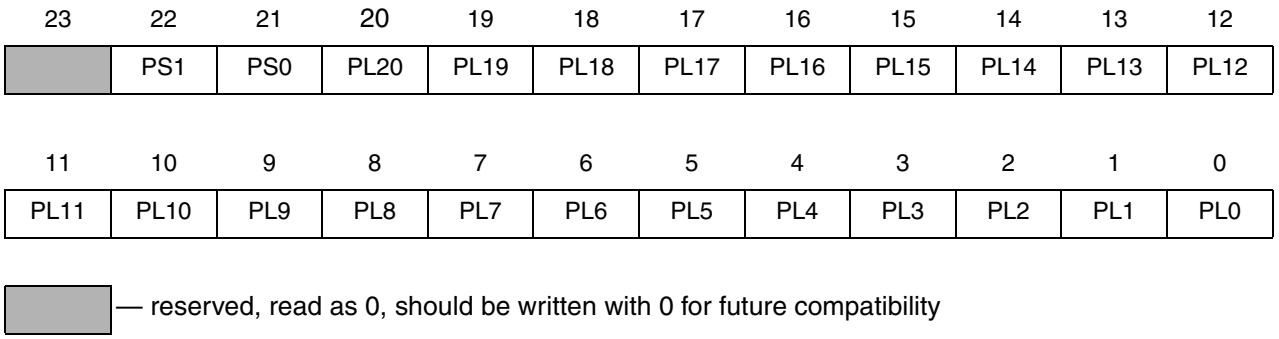

23 22 21 20 19 18 17 16 15 14 13 12
PS1 PS0 PL20 PL19 PL18 PL17 PL16 PL15 PL14 PL13 PL12
11 10 9 8 7 6 5 4 3 2 1 0
PL11 PL10 PL9 PL8 PL7 PL6 PL5 PL4 PL3 PL2 PL1 PL0
— reserved, read as 0, should be written with 0 for future compatibility

Figure 11-4 Timer Prescaler Load Register (TPLR)

11.3.2.1 TPLR Prescaler Preload Value PL[20:0] Bits 20–0

These 21 bits contain the prescaler preload value. This value is loaded into the prescaler counter when the counter value reaches zero or the counter switches state from disabled to enabled.

If PL[20:0] = N, then the prescaler counts N + 1 source clock cycles before generating a prescaler clock pulse. Therefore, the prescaler divide factor = (preload value) + 1.

The PL[20:0] bits are cleared by the hardware $\overline{\text{RESET}}$ signal or the software RESET instruction.

11.3.2.2 TPLR Prescaler Source PS[1:0] Bits 22-21

The two prescaler source (PS) bits control the source of the prescaler clock. **Table 9-1** summarizes PS bit functionality. The prescaler's use of the TIO0 signal is not affected by the TCSR settings of timer 0.

If the prescaler source clock is external, the prescaler counter is incremented by signal transitions on the TIO0 signal. The external clock is internally synchronized to the internal clock. The external clock frequency must be lower than the DSP56366 internal operating frequency divided by 4 (CLK/4).

The PS[1:0] bits are cleared by the hardware $\overline{\text{RESET}}$ signal or the software RESET instruction.

NOTE

To ensure proper operation, change the PS[1:0] bits only when the prescaler counter is disabled. Disable the prescaler counter by clearing the TE bit in the TCSR of each of three timers.

Table 11-1 Prescaler Source Selection

PS1	PS0	PRESCALER CLOCK SOURCE
0	0	Internal CLK/2
0	1	TIO0
1	0	Reserved
1	1	Reserved

11.3.2.3 TPLR Reserved Bit 23

This reserved bit is read as zero and should be written with zero for future compatibility.

11.3.3 Timer Prescaler Count Register (TPCR)

The TPCR is a 24-bit read-only register that reflects the current value in the prescaler counter. See Figure 11-5.

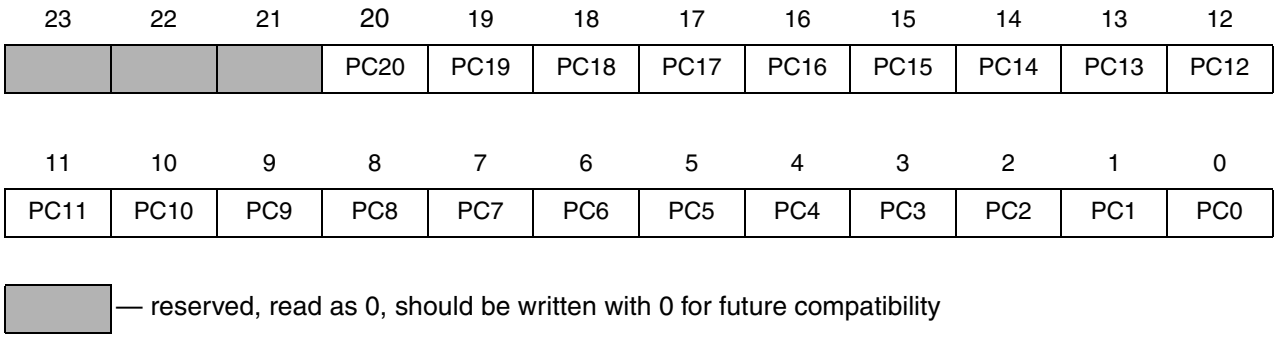

23 22 21 20 19 18 17 16 15 14 13 12
PC20 PC19 PC18 PC17 PC16 PC15 PC14 PC13 PC12
11 10 9 8 7 6 5 4 3 2 1 0
PC11 PC10 PC9 PC8 PC7 PC6 PC5 PC4 PC3 PC2 PC1 PC0
— reserved, read as 0, should be written with 0 for future compatibility

Figure 11-5 Time Prescaler Count Register (TPCR)

11.3.3.1 TPCR Prescaler Counter Value PC[20:0] Bits 20–0

These 21 bits contain the current value of the prescaler counter.

11.3.3.2 TPCR Reserved Bits 23–21

These reserved bits are read as zero and should be written with zero for future compatibility.

11.3.4 Timer Control/Status Register (TCSR)

The TCSR is a 24-bit read/write register controlling the timer and reflecting its status.

11.3.4.1 TCSR Timer Enable (TE) Bit 0

The timer enable (TE) bit is used to enable or disable the timer. Setting TE enables the timer and clears the timer counter. The counter starts counting according to the mode selected by the timer control (TC[3:0]) bit values.

Clearing the TE bit disables the timer. The TE bit is cleared by the hardware $\overline{\text{RESET}}$ signal or the software RESET instruction.

NOTE

When timer 0 is disabled and TIO0 is not in GPIO mode, the pin is tri-stated. To prevent undesired spikes on TIO0 when Timer 0 is switched from tri-state to an active state, TIO0 should be tied to the power supply with a pullup or pulldown resistor.

11.3.4.2 TCSR Timer Overflow Interrupt Enable (TOIE) Bit 1

The TOIE bit is used to enable the timer overflow interrupts. Setting TOIE enables overflow interrupt generation. The timer counter can hold a maximum value of $FFFFFF. When the counter value is at the maximum value and a new event causes the counter to be incremented to $000000, the timer generates an overflow interrupt.

Clearing the TOIE bit disables overflow interrupt generation. The TOIE bit is cleared by the hardware $\overline{\text{RESET}}$ signal or the software RESET instruction.

11.3.4.3 TCSR Timer Compare Interrupt Enable (TCIE) Bit 2

The Timer Compare Interrupt Enable (TCIE) bit is used to enable or disable the timer compare interrupts. Setting TCIE enables the compare interrupts. In the timer, PWM, or watchdog modes, a compare interrupt is generated after the counter value matches the value of the TCPR. The counter will start counting up from the number loaded from the TLR and if the TCPR value is N, an interrupt occurs after (N – M + 1) events, where M is the value of TLR.

Clearing the TCIE bit disables the compare interrupts. The TCIE bit is cleared by the hardware $\overline{\text{RESET}}$ signal or the software RESET instruction.

11.3.4.4 TCSR Timer Control (TC[3:0]) Bits 4–7

The four TC bits control the source of the timer clock, the behavior of the TIO0 signal, and the timer mode of operation. Table 11-2 summarizes the TC bit functionality. A detailed description of the timer operating modes is given in Section 11.4, "Timer Modes of Operation".

The TC bits are cleared by the hardware $\overline{\text{RESET}}$ signal or the software RESET instruction.

NOTE

If the clock is external, the counter is incremented by the transitions on the TIO0 signal. The external clock is internally synchronized to the internal clock, and its frequency should be lower than the internal operating frequency divided by 4 (CLK/4).

To ensure proper operation, the TC[3:0] bits should be changed only when the timer is disabled (when the TE bit in the TCSR has been cleared).

Table 11-2 Timer Control Bits for Timer 0

Bit Settings				Mode Characteristics			
TC3	TC2	TC1	TC0	Mode Number	Mode Function	TIO0	Clock
0	0	0	0	0	Timer and GPIO	GPIO[1]	Internal
0	0	0	1	1	Timer pulse	Output	Internal
0	0	1	0	2	Timer toggle	Output	Internal
0	0	1	1	3	Event counter	Input	External
0	1	0	0	4	Input width measurement	Input	Internal
0	1	0	1	5	Input period measurement	Input	Internal
0	1	1	0	6	Capture event	Input	Internal
0	1	1	1	7	Pulse width modulation	Output	Internal
1	0	0	0	8	*Reserved*	—	—
1	0	0	1	9	Watchdog pulse	Output	Internal
1	0	1	0	10	Watchdog toggle	Output	Internal
1	0	1	1	11	*Reserved*	—	—
1	1	0	0	12	*Reserved*	—	—
1	1	0	1	13	*Reserved*	—	—
1	1	1	0	14	*Reserved*	—	—
1	1	1	1	15	*Reserved*	—	—

[1] The GPIO function is enabled only if all of the TC[3:0] bits are zero.

Table 11-3 Timer Control Bits for Timers 1 and 2

TC3	TC2	TC1	TC0	Clock	Mode
0	0	0	0	Internal	Timer
0	0	0	1	—	*Reserved*
0	0	1	X	—	*Reserved*
0	1	X	X	—	*Reserved*
1	X	X	X	—	*Reserved*

11.3.4.5 TCSR Inverter (INV) Bit 8

The INV bit affects the polarity of the incoming signal on the TIO0 input signal and the polarity of the output pulse generated on the TIO0 output signal. The effects of the INV bit are summarized in Table 11-4.

This bit is not in use for timers 1 and 2. It should be left cleared.

Table 11-4 Inverter (INV) Bit Operation

Mode	TIO0 Programmed as Input		TIO0 Programmed as Output	
	INV = 0	INV = 1	INV = 0	INV = 1
0	GPIO signal on the TIO0 signal read directly	GPIO signal on the TIO0 signal inverted	Bit written to GPIO put on TIO0 signal directly	Bit written to GPIO inverted and put on TIO0 signal
1	Counter is incremented on the **rising** edge of the signal from the TIO0 signal	Counter is incremented on the **falling** edge of the signal from the TIO0 signal	—	—
2	Counter is incremented on the **rising** edge of the signal from the TIO0 signal	Counter is incremented on the **falling** edge of the signal from the TIO0 signal	TCRx output put on TIO0 signal directly	TCRx output inverted and put on TIO0 signal
3	Counter is incremented on the **rising** edge of the signal from the TIO0 signal	Counter is incremented on the **falling** edge of the signal from the TIO0 signal	—	—
4	Width of the **high** input pulse is measured.	Width of the **low** input pulse is measured.	—	—
5	Period is measured between the **rising** edges of the input signal.	Period is measured between the **falling** edges of the input signal.	—	—
6	Event is captured on the **rising** edge of the signal from the TIO0 signal	Event is captured on the **falling** edge of the signal from the TIO0 signal	—	—
7	—	—	Pulse generated by the timer has **positive** polarity	Pulse generated by the timer has **negative** polarity
9	—	—	Pulse generated by the timer has **positive** polarity	Pulse generated by the timer has **negative** polarity
10	—	—	Pulse generated by the timer has **positive** polarity	Pulse generated by the timer has **negative** polarity

The INV bit is cleared by the hardware $\overline{\text{RESET}}$ signal or the software RESET instruction.

NOTE

The INV bit affects both the timer and GPIO modes of operation. To ensure correct operation, this bit should be changed only when one or both of the following conditions is true:

- The timer has been disabled by clearing the TE bit in the TCSR.
- The timer is in GPIO mode.

The INV bit does not affect the polarity of the prescaler source when TIO0 is used as input to the prescaler.

11.3.4.6 TCSR Timer Reload Mode (TRM) Bit 9

The TRM bit controls the counter preload operation.

In timer (0–3) and watchdog (9–10) modes, the counter is preloaded with the TLR value after the TE bit is set and the first internal or external clock signal is received. If the TRM bit is set, the counter is reloaded each time after it reaches the value contained by the TCR. In PWM mode (7), the counter is reloaded each time counter overflow occurs. In measurement (4–5) modes, if the TRM and the TE bits are set, the counter is preloaded with the TLR value on each appropriate edge of the input signal.

If the TRM bit is cleared, the counter operates as a free-running counter and is incremented on each incoming event. The TRM bit is cleared by the hardware $\overline{\text{RESET}}$ signal or the software RESET instruction.

11.3.4.7 TCSR Direction (DIR) Bit 11

The DIR bit determines the behavior of the TIO0 signal when it is used as a GPIO pin. When the DIR bit is set, the TIO0 signal is an output; when the DIR bit is cleared, the TIO0 signal is an input. The TIO0 signal can be used as a GPIO only when the TC[3:0] bits are all cleared. If any of the TC[3:0] bits are set, then the GPIO mode is disabled and the DIR bit has no effect.

The DIR bit is cleared by the hardware $\overline{\text{RESET}}$ signal or the software RESET instruction.

This bit is not in use for timers 1 and 2. It should be left cleared.

11.3.4.8 TCSR Data Input (DI) Bit 12

The DI bit reflects the value of the TIO0 signal. If the INV bit is set, the value of the TIO0 signal is inverted before it is written to the DI bit. If the INV bit is cleared, the value of the TIO0 signal is written directly to the DI bit.

DI is cleared by the hardware $\overline{\text{RESET}}$ signal or the software RESET instruction.

11.3.4.9 TCSR Data Output (DO) Bit 13

The DO bit is the source of the TIO0 value when it is a data output signal. The TIO0 signal is data output when the GPIO mode is enabled and DIR is set. A value written to the DO bit is written to the TIO0 signal. If the INV bit is set, the value of the DO bit is inverted when written to the TIO0 signal. When the INV bit is cleared, the value of the DO bit is written directly to the TIO0 signal. When GPIO mode is disabled, writing the DO bit has no effect.

The DO bit is cleared by the hardware $\overline{\text{RESET}}$ signal or the software RESET instruction.

This bit is not in use for timers 1 and 2. It should be left cleared.

11.3.4.10 TCSR Prescaler Clock Enable (PCE) Bit 15

The PCE bit is used to select the prescaler clock as the timer source clock. When the PCE bit is cleared, the timer uses either an internal (CLK/2) signal or an external signal (TIO0) as its source clock. When the PCE bit is set, the prescaler output is used as the timer source clock for the counter regardless of the timer operating mode. To ensure proper operation, the PCE bit should be changed only when the timer is disabled (when the TE bit is cleared). Which source clock is used for the prescaler is determined by the PS[1:0] bits of the TPLR. Timers 1 and 2 can be clocked by the prescaler clock derived from TIO0.

11.3.4.11 TCSR Timer Overflow Flag (TOF) Bit 20

The TOF bit is set to indicate that counter overflow has occurred. This bit is cleared by writing a 1 to the TOF bit. Writing a 0 to the TOF bit has no effect. The bit is also cleared when the timer overflow interrupt is serviced.

The TOF bit is cleared by the hardware $\overline{\text{RESET}}$ signal, the software RESET instruction, the STOP instruction, or by clearing the TE bit to disable the timer.

11.3.4.12 TCSR Timer Compare Flag (TCF) Bit 21

The TCF bit is set to indicate that the event count is complete. In the timer, PWM, and watchdog modes, the TCF bit is set when (N – M + 1) events have been counted (N is the value in the compare register and M is the TLR value). In the measurement modes, the TCF bit is set when the measurement has been completed.

The TCF bit is cleared by writing a one into the TCF bit. Writing a zero into the TCF bit has no effect. The bit is also cleared when the timer compare interrupt is serviced.

The TCF bit is cleared by the hardware $\overline{\text{RESET}}$ signal, the software RESET instruction, the STOP instruction, or by clearing the TE bit to disable the timer.

NOTE

The TOF and TCF bits are cleared by writing a one to the specific bit. In order to assure that only the desired bit is cleared, do not use the BSET command. The proper way to clear these bits is to write (using a MOVEP instruction) a one to the flag to be cleared and a zero to the other flag.

11.3.4.13 TCSR Reserved Bits (Bits 3, 10, 14, 16-19, 22, 23)

These reserved bits are read as zero and should be written with zero for future compatibility.

11.3.5 Timer Load Register (TLR)

The TLR is a 24-bit write-only register. In all modes, the counter is preloaded with the TLR value after the TE bit in the TCSR is set and a first event occurs. The programmer must initialize the TLR to ensure correct operation in the appropriate timer operating modes.

- In timer modes, if the timer reload mode (TRM) bit in the TCSR is set, the counter is reloaded each time after it has reached the value contained by the timer compare register (TCR) and the new event occurs.
- In measurement modes, if the TRM bit in the TCSR is set and the TE bit in the TCSR is set, the counter is reloaded with the value in the TLR on each appropriate edge of the input signal.
- In PWM modes, if the TRM bit in the TCSR is set, the counter is reloaded each time after it has overflowed and the new event occurs.
- In watchdog modes, if the TRM bit in the TCSR is set, the counter is reloaded each time after it has reached the value contained by the TCR and the new event occurs. In this mode, the counter is also reloaded whenever the TLR is written with a new value while the TE bit in the TCSR is set.
- In all modes, if the TRM bit in the TCSR is cleared (TRM = 0), the counter operates as a free-running counter.

11.3.6 Timer Compare Register (TCPR)

The TCPR is a 24-bit read/write register that contains the value to be compared to the counter value. These two values are compared every timer clock after the TE bit in the TCSR is set. When the values match, the timer compare flag (TCF) bit is set and an interrupt is generated if interrupts are enabled (if the timer compare interrupt enable (TCIE) bit in the TCSR is set). The programmer must initialize the TCPR to ensure correct operation in the appropriate timer operating modes. The TCPR is ignored in measurement modes.

11.3.7 Timer Count Register (TCR)

The TCR is a 24-bit read-only register. In timer and watchdog modes, the counter's contents can be read at any time by reading the TCR register. In measurement modes, the TCR is loaded with the current value of the counter on the appropriate edge of the input signal, and its value can be read to determine the width, period, or delay of the leading edge of the input signal. When the timer is in measurement modes, the TIO0 signal is used for the input signal.

11.4 Timer Modes of Operation

Each timer has various operational modes that meet a variety of system requirements. These modes are as follows:

- Timer
  - GPIO, mode 0: Internal timer interrupt generated by the internal clock
  - Pulse, mode 1: External timer pulse generated by the internal clock
  - Toggle, mode 2: Output timing signal toggled by the internal clock

  - Event counter, mode 3: Internal timer interrupt generated by an external clock
- Measurement
  - Input width, mode 4: Input pulse width measurement
  - Input pulse, mode 5: Input signal period measurement
  - Capture, mode 6: Capture external signal
- PWM, mode 7: Pulse Width Modulation
- Watchdog
  - Pulse, mode 9: Output pulse, internal clock
  - Toggle, mode 10: Output toggle, internal clock

These modes are described in detail below. Timer modes are selected by setting the TC[3:0] bits in the TCSR. Table 11-2 and Table 11-3 show how the different timer modes are selected by setting the bits in the TCSR. Table 11-2 also shows the TIO0 signal direction and the clock source for each timer mode.

NOTE

To ensure proper operation, the TC[3:0] bits should be changed only when the timer is disabled (i.e., when the TE bit in the TCSR is cleared).

11.4.1 Timer Modes

11.4.1.1 Timer GPIO (Mode 0)

Bit Settings				Mode Characteristics				
TC3	TC2	TC1	TC0	TIO0	Clock	#	KIND	NAME
0	0	0	0	GPIO	Internal	0	Timer	GPIO

In this mode, the timer generates an internal interrupt when a counter value is reached (if the timer compare interrupt is enabled). Note that any of the three timers can be placed in GPIO mode to generate internal interrupts, but only timer 0 provides actual external GPIO access on the TIO0 signal.

Set the TE bit to clear the counter and enable the timer. Load the value the timer is to count into the TCPR. The counter is loaded with the TLR value when the first timer clock signal is received. The timer clock can be taken from either the DSP56366 clock divided by two (CLK/2) or from the prescaler clock output. Each subsequent clock signal increments the counter.

When the counter equals the TCPR value, the TCF bit in TCSR is set, and a compare interrupt is generated if the TCIE bit is set. If the TRM bit in the TCSR is set, the counter is reloaded with the TLR value at the next timer clock and the count is resumed. If the TRM bit is cleared, the counter continues to be incremented on each timer clock signal.

This process is repeated until the timer is disabled (i.e., TE is cleared).

If the counter overflows, the TOF bit is set, and if TOIE is set, an overflow interrupt is generated.

The counter contents can be read at any time by reading the TCR.

11.4.1.2 Timer Pulse (Mode 1)

Bit Settings				Mode Characteristics				
TC3	TC2	TC1	TC0	TIO0	Clock	#	KIND	NAME
0	0	0	1	Output	Internal	1	Timer	Pulse

In this mode, the timer generates a compare interrupt when the timer count reaches a preset value. In addition, timer 0 provides an external pulse on its TIO0 signal.

Set the TE bit to clear the counter and enable the timer. The value to which the timer is to count is loaded into the TCPR. The counter is loaded with the TLR value when the first timer clock signal is received. The TIO0 signal is loaded with the value of the INV bit. The timer clock signal can be taken from either the DSP56366 clock divided by two (CLK/2) or from the prescaler clock output. Each subsequent clock signal increments the counter.

When the counter matches the TCPR value, the TCF bit in TCSR is set and a compare interrupt is generated if the TCIE bit is set. The polarity of the TIO0 signal is inverted for one timer clock period.

If the TRM bit is set, the counter is loaded with the TLR value on the next timer clock and the count is resumed. If the TRM bit is cleared, the counter continues to be incremented on each timer clock.

This process is repeated until the TE bit is cleared (disabling the timer).

The value of the TLR sets the delay between starting the timer and the generation of the output pulse. To generate successive output pulses with a delay of X clocks between signals, the TLR value should be set to X/2 and the TRM bit should be set.

This process is repeated until the timer is disabled (i.e., TE is cleared).

If the counter overflows, the TOF bit is set, and if TOIE is set, an overflow interrupt is generated.

The counter contents can be read at any time by reading the TCR.

11.4.1.3 Timer Toggle (Mode 2)

Bit Settings				Mode Characteristics				
TC3	TC2	TC1	TC0	TIO0	Clock	#	KIND	NAME
0	0	1	0	Output	Internal	0	Timer	Toggle

In this mode, the timer generates a periodic interrupt; timer 0 also toggles the polarity of the TIO0 signal.

Set the TE bit in the TCR to clear the counter and enable the timer. The value the timer is to count is loaded into the TPCR. The counter is loaded with the TLR value when the first timer clock signal is received. The TIO0 signal is loaded with the value of the INV bit. The timer clock signal can be taken from either the DSP56366 clock divided by two (CLK/2) or from the prescaler clock output. Each subsequent clock signal increments the counter.

When the counter value matches the value in the TCPR, the polarity of the TIO0 output signal is inverted. The TCF bit in the TCSR is set and a compare interrupt is generated if the TCIE bit is set.

If the TRM bit is set, the counter is loaded with the value of the TLR when the next timer clock is received, and the count is resumed. If the TRM bit is cleared, the counter continues to be incremented on each timer clock.

This process is repeated until the TE bit is cleared, disabling the timer.

The TLR value in the TCPR sets the delay between starting the timer and toggling the TIO0 signal. To generate output signals with a delay of X clock cycles between toggles, the TLR value should be set to X/2 and the TRM bit should be set.

This process is repeated until the timer is disabled (i.e., TE is cleared). If the counter overflows, the TOF bit is set, and if TOIE is set, an overflow interrupt is generated.

The counter contents can be read at any time by reading the TCR.

11.4.1.4 Timer Event Counter (Mode 3)

Bit Settings				Mode Characteristics				
TC3	TC2	TC1	TC0	TIO0	Clock	#	KIND	NAME
0	0	1	1	Input	External	3	Timer	Event Counter

In this mode, the timer counts internal events and issues an interrupt when a preset number of events is counted. Timer 0 can also count external events.

Set the TE bit to clear the counter and enable the timer. The number of events the timer is to count is loaded into the TPCR. The counter is loaded with the TLR value when the first timer clock signal is received. The timer clock signal is provided by the prescaler clock output. Timer 0 can be also be clocked from the TIO0 input signal. Each subsequent clock signal increments the counter. If an external clock is used, it must be internally synchronized to the internal clock and its frequency must be less than the DSP56366 internal operating frequency divided by 4.

The value of the INV bit in the TCSR determines whether low-to-high (0 to 1) transitions or high-to-low (1 to 0) transitions increment the counter. If the INV bit is set, high-to-low transitions increment the counter. If the INV bit is cleared, low-to-high transitions increment the counter.

When the counter matches the value contained in the TCPR, the TCF bit in the TCSR is set and a compare interrupt is generated if the TCIE bit is set. If the TRM bit is set, the counter is loaded with the value of the TLR when the next timer clock is received, and the count is resumed. If TRM bit is cleared, the counter continues to be incremented on each timer clock.

This process is repeated until the timer is disabled (i.e., TE is cleared). If the counter overflows, the TOF bit is set, and if TOIE is set, an overflow interrupt is generated.

The counter contents can be read at any time by reading the TCR.

11.4.2 Signal Measurement Modes

The following signal measurement modes are provided:

- Measurement input width
- Measurement input period
- Measurement capture

These functions are available only on timer 0.

11.4.2.1 Measurement Accuracy

The external signal is synchronized with the internal clock used to increment the counter. This synchronization process can cause the number of clocks measured for the selected signal value to vary from the actual signal value by plus or minus one counter clock cycle.

11.4.2.2 Measurement Input Width (Mode 4)

Bit Settings				Mode Characteristics				
TC3	TC2	TC1	TC0	Mode	Name	Kind	TIO0	Clock
0	1	0	0	4	Input Width	Measurement	Input	Internal

In this mode, the timer 0 counts the number of clocks that occur between opposite edges of an input signal.

Set the TE bit to clear the counter and enable the timer. Load the timer's count value into the TLR. After the first appropriate transition (as determined by the INV bit) occurs on the TIO0 input pin, the counter is loaded with the TLR value on the first timer clock signal received either from the DSP56366 clock divided by two (CLK/2) or from the prescaler clock input. Each subsequent clock signal increments the counter.

If the INV bit is set, the timer starts on the first high-to-low (1 to 0) signal transition on the TIO0 signal. If the INV bit is cleared, the timer starts on the first low-to-high (0 to 1) transition on the TIO0 signal.

When the first transition opposite in polarity to the INV bit setting occurs on the TIO0 signal, the counter stops. The TCF bit in the TCSR is set and a compare interrupt is generated if the TCIE bit is set. The value of the counter (which measures the width of the TIO0 pulse) is loaded into the TCR. The TCR can be read to determine the external signal pulse width.

If the TRM bit is set, the counter is loaded with the TLR value on the first timer clock received following the next valid transition occurring on the TIO0 input pin and the count is resumed. If the TRM bit is cleared, the counter continues to be incremented on each timer clock.

This process is repeated until the timer is disabled (i.e., TE is cleared).

If the counter overflows, the TOF bit is set, and if TOIE is set, an overflow interrupt is generated.

The counter contents can be read at any time by reading the TCR.

11.4.2.3 Measurement Input Period (Mode 5)

Bit Settings				Mode Characteristics				
TC3	TC2	TC1	TC0	Mode	Name	Kind	TIO0	Clock
0	1	0	1	5	Input Period	Measurement	Input	Internal

In this mode, the timer counts the period between the reception of signal edges of the same polarity across the TIO0 signal.

Set the TE bit to clear the counter and enable the timer. The value the timer is to count is loaded into the TLR. The value of the INV bit determines whether the period is measured between consecutive low-to-high (0 to 1) transitions of TIO0 or between consecutive high-to-low (1 to 0) transitions of TIO0. If INV is set, high-to-low signal transitions are selected. If INV is cleared, low-to-high signal transitions are selected.

After the first appropriate transition occurs on the TIO0 input pin, the counter is loaded with the TLR value on the first timer clock signal received from either the DSP56366 clock divided by two (CLK/2) or the prescaler clock output. Each subsequent clock signal increments the counter.

On the next signal transition of the same polarity that occurs on TIO0, the TCF bit in the TCSR is set and a compare interrupt is generated if the TCIE bit is set. The contents of the counter are loaded into the TCR. The TCR then contains the value of the time that elapsed between the two signal transitions on the TIO0 signal.

After the second signal transition, if the TRM bit is set, the TE bit is set to clear the counter and enable the timer. The counter is repeatedly loaded and incremented until the timer is disabled. If the TRM bit is cleared, the counter continues to be incremented until it overflows.

This process is repeated until the timer is disabled (i.e., TE is cleared).

If the counter overflows, the TOF bit is set, and if TOIE is set, an overflow interrupt is generated.

The counter contents can be read at any time by reading the TCR.

11.4.2.4 Measurement Capture (Mode 6)

Bit Settings				Mode Characteristics				
TC3	TC2	TC1	TC0	Mode	Name	Kind	TIO0	Clock
0	1	1	0	6	Capture	Measurement	Input	Internal

In this mode, the timer counts the number of clocks that elapse between starting the timer and receiving an external signal.

Set the TE bit to clear the counter and enable the timer. The value the timer is to count is loaded into the TLR. When the first timer clock signal is received, the counter is loaded with the TLR value. The timer

clock signal can be taken from either the DSP56366 clock divided by two (CLK/2) or from the prescaler clock output. Each subsequent clock signal increments the counter.

At the first appropriate transition of the external clock detected on the TIO0 signal, the TCF bit in the TCSR is set and, if the TCIE bit is set, a compare interrupt is generated. The counter halts. The contents of the counter are loaded into the TCR. The value of the TCR represents the delay between the setting of the TE bit and the detection of the first clock edge signal on the TIO0 signal.

If the INV bit is set, a high-to-low transition signals the end of the timing period. If INV is cleared, a low-to-high transition signals the end of the timing period.

If the counter overflows, the TOF bit is set, and if TOIE is set, an overflow interrupt is generated.

The counter contents can be read at any time by reading the TCR.

11.4.3 Pulse Width Modulation (PWM, Mode 7)

Bit Settings				Mode Characteristics				
TC3	TC2	TC1	TC0	Mode	Name	Kind	TIO0	Clock
0	1	1	1	7	Pulse Width Modulation	PWM	Output	Internal

In this mode, the timer generates periodic pulses of a preset width. This function is available only on timer 0.

Set the TE bit to clear the counter and enable the timer. The value the timer is to count is loaded into the TPCR. When first timer clock is received from either the DSP56366 internal clock divided by two (CLK/2) or the prescaler clock output, the counter is loaded with the TLR value. Each subsequent timer clock increments the counter.

When the counter equals the value in the TCPR, the TIO0 output pin is toggled and the TCF bit in the TCSR is set. The contents of the counter are placed into the TCR. If the TCIE bit is set, a compare interrupt is generated. The counter continues to be incremented on each timer clock.

If counter overflow has occurred, the TIO0 output pin is toggled, the TOF bit in TCSR is set, and an overflow interrupt is generated if the TOIE bit is set. If the TRM bit is set, the counter is loaded with the TLR value on the next timer clock and the count is resumed. If the TRM bit is cleared, the counter continues to be incremented on each timer clock.

This process is repeated until the timer is disabled by clearing the TE bit.

TIO0 signal polarity is determined by the value of the INV bit. When the counter is started by setting the TE bit, the TIO0 signal assumes the value of the INV bit. On each subsequent toggling of the TIO0 signal, the polarity of the TIO0 signal is reversed. For example, if the INV bit is set, the TIO0 signal generates the following signal: 1010. If the INV bit is cleared, the TIO0 signal generates the following signal: 0101.

The counter contents can be read at any time by reading the TCR.

The value of the TLR determines the output period ($FFFFFF – TLR + 1). The timer counter increments the initial TLR value and toggles the TIO0 signal when the counter value exceeds $FFFFFF.

The duty cycle of the TIO0 signal is determined by the value in the TCPR. When the value in the TLR is incremented to a value equal to the value in the TCPR, the TIO0 signal is toggled. The duty cycle is equal to ($FFFFFF – TCPR) divided by ($FFFFFF – TLR + 1). For a 50% duty cycle, the value of TCPR is equal to ($FFFFFF + TLR + 1) / 2.

NOTE

The value in TCPR must be greater than the value in TLR.

11.4.4 Watchdog Modes

11.4.4.1 Watchdog Pulse (Mode 9)

Bit Settings				Mode Characteristics				
TC3	TC2	TC1	TC0	Mode	Name	Kind	TIO0	Clock
1	0	0	1	9	Pulse	Watchdog	Output	Internal

In this mode, the timer generates an interrupt at a preset rate. Timer 0 also generates pulse on TIO0. The signal period is equal to the period of one timer clock.

Set the TE bit to clear the counter and enable the timer. The value the timer is to count is loaded into the TCPR. The counter is loaded with the TLR value on the first timer clock received from either the DSP56366 internal clock divided by two (CLK/2) or the prescaler clock output. Each subsequent timer clock increments the counter.

When the counter matches the value of the TCPR, the TCF bit in the TCSR is set and a compare interrupt is generated if the TCIE bit is also set.

If the TRM bit is set, the counter is loaded with the TLR value on the next timer clock and the count is resumed. If the TRM bit is cleared, the counter continues to be incremented on each subsequent timer clock.

This process is repeated until the timer is disabled (i.e., TE is cleared).

If the counter overflows, the TOF bit is set, and if TOIE is set, an overflow interrupt is generated. Timer 0 also generates an output pulse on the TIO0 signal with a pulse width equal to the timer clock period. The pulse polarity is determined by the value of the INV bit. If the INV bit is set, the pulse polarity is high (logical 1). If the INV bit is cleared, the pulse polarity is low (logical 0).

The counter contents can be read at any time by reading the TCR.

The counter is reloaded whenever the TLR is written with a new value while the TE bit is set.

NOTE

In this mode, internal logic preserves the TIO0 value and direction for an additional 2.5 internal clock cycles after the DSP56366 hardware $\overline{\text{RESET}}$ signal is asserted. This ensures that a valid $\overline{\text{RESET}}$ signal is generated when the TIO0 signal is used to reset the DSP56366.

11.4.4.2 Watchdog Toggle (Mode 10)

Bit Settings				Mode Characteristics				
TC3	TC2	TC1	TC0	Mode	NAME	Kind	TIO0	Clock
1	0	1	0	10	Toggle	Watchdog	Output	Internal

In this mode, the timer generates an interrupt at a preset rate. Timer 0 also toggles the output on TIO0.

Set the TE bit to clear the counter and enable the timer. The value the timer is to count is loaded into the TPCR. The counter is loaded with the TLR value on the first timer clock received from either the DSP56366 internal clock divided by two (CLK/2) or the prescaler clock output. Each subsequent timer clock increments the counter. The TIO0 signal is set to the value of the INV bit.

When the counter equals the value in the TCPR, the TCF bit in the TCSR is set, and a compare interrupt is generated if the TCIE bit is also set. If the TRM bit is set, the counter is loaded with the TLR value on the next timer clock and the count is resumed. If the TRM bit is cleared, the counter continues to be incremented on each subsequent timer clock

When counter overflow has occurred, the polarity of the TIO0 output pin is inverted, the TOF bit in the TCSR is set, and an overflow interrupt is generated if the TOIE bit is also set. The TIO0 polarity is determined by the INV bit.

The counter is reloaded whenever the TLR is written with a new value while the TE bit is set. This process is repeated until the timer is disabled by clearing the TE bit. The counter contents can be read at any time by reading the TCR register.

NOTE

In this mode, internal logic preserves the TIO0 value and direction for an additional 2.5 internal clock cycles after the DSP56366 hardware $\overline{\text{RESET}}$ signal is asserted. This ensures that a valid $\overline{\text{RESET}}$ signal is generated when the TIO0 signal is used to reset the DSP56366.

11.4.5 Reserved Modes

Modes 8, 11, 12, 13, 14, and 15 are reserved.

11.4.6 Special Cases

The following special cases apply during wait and stop state.

11.4.6.1 Timer Behavior during Wait

Timer clocks are active during the execution of the WAIT instruction and timer activity is undisturbed. If a timer interrupt is generated, the DSP56366 leaves the wait state and services the interrupt.

11.4.6.2 Timer Behavior during Stop

During the execution of the STOP instruction, the timer clocks are disabled, timer activity is stopped, and the TIO0 signal is disconnected. Any external changes that happen to the TIO0 signal is ignored when the DSP56366 is the stop state. To ensure correct operation, the timers should be disabled before the DSP56366 is placed into the stop state.

11.4.7 DMA Trigger

Each timer can also be used to trigger DMA transfers. For this to occur, a DMA channel must be programmed to be triggered by a timer event. The timer issues a DMA trigger on every event in all modes of operation. The DMA channel does not have the capability to save multiple DMA triggers generated by the timer. To ensure that all DMA triggers are serviced, the user must provide for the preceding DMA trigger to be serviced before the next trigger is received by the DMA channel.

NOTES

Appendix A Bootstrap ROM Contents

A.1 DSP56366 Bootstrap Program

```
; BOOTSTRAP CODE FOR DSP56366 Rev. 0 silicon - (C) Copyright 1999 Motorola Inc.
;
;
; Revision 0.0 1999/JAN/26 - Modified from 56362_RevA_regular_boot_rev01.asm:
;                          - Change the length of xram and the length of yram
;                                  in burn-in code
;                          - Change the address of the reserved area in the
;                                  Program ROM to $FFAF80 - $FFAFFF
;
; Revision 0.1 1999/MAR/29 - Enabled 100ns I2C filter in bootstrap
;                            mode 0110.
;                          - Added 5 NOP instructions after OnCE enable.
;
; This is the Bootstrap program contained in the DSP56366 192-word Boot
; ROM. This program can load any program RAM segment from an external
; EPROM, from the Host Interface or from the SHI serial interface.
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; If MD:MC:MB:MA=x000, then the Boot ROM is bypassed and the DSP56366
; will start fetching instructions beginning with address $C00000 (MD=0)
; or $008000 (MD=1) assuming that an external memory of SRAM type is
; used. The accesses will be performed using 31 wait states with no
; address attributes selected (default area).
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; If MD:MC:MB:MA=0001, then it loads a program RAM segment from consecutive
; byte-wide P memory locations, starting at P:$D00000 (bits 7-0).
; The memory is selected by the Address Attribute AA1 and is accessed with
; 31 wait states.
; The EPROM bootstrap code expects to read 3 bytes
; specifying the number of program words, 3 bytes specifying the address
; to start loading the program words and then 3 bytes for each program
; word to be loaded. The number of words, the starting address and the
; program words are read least significant byte first followed by the
; mid and then by the most significant byte.
; The program words will be condensed into 24-bit words and stored in
; contiguous PRAM memory locations starting at the specified starting address.
; After reading the program words, program execution starts from the same
; address where loading started.
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; If MD:MC:MB:MA=0010, then the bootstrap code jumps to the internal
```

```
; Program ROM, without loading the Program RAM.
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Operation mode MD:MC:MB:MA=0011 is reserved.
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; If MD:MC:MB:MA=01xx, then the Program RAM is loaded from the SHI.
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Operation mode MD:MC:MB:MA=1001 is used for burn-in testing.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Operation mode MD:MC:MB:MA=1010 is reserved
; Operation mode MD:MC:MB:MA=1011 is reserved
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; If MD:MC:MB:MA=1100, then it loads the program RAM from the Host
; Interface programmed to operate in the ISA mode.
; The HOST ISA bootstrap code expects to read a 24-bit word
; specifying the number of program words, a 24-bit word specifying the address
; to start loading the program words and then a 24-bit word for each program
; word to be loaded. The program words will be stored in
; contiguous PRAM memory locations starting at the specified starting address.
; After reading the program words, program execution starts from the same
; address where loading started.
; The Host Interface bootstrap load program may be stopped by
; setting the Host Flag 0 (HF0). This will start execution of the loaded
; program from the specified starting address.
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; If MD:MC:MB:MA=1101, then it loads the program RAM from the Host
; Interface programmed to operate in the HC11 non multiplexed mode.
;
; The HOST HC11 bootstrap code expects to read a 24-bit word
; specifying the number of program words, a 24-bit word specifying the address
; to start loading the program words and then a 24-bit word for each program
; word to be loaded. The program words will be stored in
; contiguous PRAM memory locations starting at the specified starting address.
; After reading the program words, program execution starts from the same
; address where loading started.
; The Host Interface bootstrap load program may be stopped by
; setting the Host Flag 0 (HF0). This will start execution of the loaded
; program from the specified starting address.
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; If MD:MC:MB:MA=1110, then it loads the program RAM from the Host
; Interface programmed to operate in the 8051 multiplexed bus mode,
; in double-strob pin configuration.
; The HOST 8051 bootstrap code expects accesses that are byte wide.
```

```
; The HOST 8051 bootstrap code expects to read 3 bytes forming a 24-bit word
; specifying the number of program words, 3 bytes forming a 24-bit word
; specifying the address to start loading the program words and then 3 bytes
; forming 24-bit words for each program word to be loaded.
; The program words will be stored in contiguous PRAM memory locations
; starting at the specified starting address.
; After reading the program words, program execution starts from the same
; address where loading started.
; The Host Interface bootstrap load program may be stopped by setting the
; Host Flag 0 (HF0). This will start execution of the loaded program from
; the specified starting address.
;
; The base address of the HDI08 in multiplexed mode is 0x80 and is not
; modified by the bootstrap code. All the address lines are enabled
; and should be connected accordingly.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; If MD:MC:MB:MA=1111, then it loads the program RAM from the Host
; Interface programmed to operate in the MC68302 (IMP) bus mode,
; in single-strob pin configuration.
; The HOST MC68302 bootstrap code expects accesses that are byte wide.
; The HOST MC68302 bootstrap code expects to read 3 bytes forming a 24-bit word
; specifying the number of program words, 3 bytes forming a 24-bit word
; specifying the address to start loading the program words and then 3 bytes
; forming 24-bit words for each program word to be loaded.
; The program words will be stored in contiguous PRAM memory locations
; starting at the specified starting address.
; After reading the program words, program execution starts from the same
; address where loading started.
; The Host Interface bootstrap load program may be stopped by setting the
; Host Flag 0 (HF0). This will start execution of the loaded program from
; the specified starting address.
;

        page    132,55,0,0,0

        opt     cex,mex,mu

;;
;;;;;;;;;;;;;;;;;;;; GENERAL EQUATES ;;;;;;;;;;;;;;;;;;;;;;;;
;;
BOOT    equ     $D00000         ; this is the location in P memory
                                ; on the external memory bus
                                ; where the external byte-wide
                                ; EPROM is located
AARV    equ     $D00409         ; AAR1 selects the EPROM as CE~
                                ; mapped as P from $D00000 to
                                ; $DFFFFF, active low
PROMADDR equ    $FF1000         ; Starting PROM address

MA      EQU     0
MB      EQU     1
MC      EQU     2
MD      EQU     3
```

```
;;
;;;;;;;;;;;;;;;;;;; DSP I/O REGISTERS ;;;;;;;;;;;;;;;;;;;;;;
;;

M_AAR1  EQU     $FFFFF8         ; Address Attribute Register 1

M_OGDB  EQU     $FFFFFC         ; OnCE GDB Register

M_HPCR  EQU     $FFFFC4         ; Host Polarity Control Register
M_HSR   EQU     $FFFFC3         ; Host Status Register
M_HORX  EQU     $FFFFC6         ; Host Receive Register
HRDF    EQU     $0              ; Host Receive Data Full
HF0     EQU     $3              ; Host Flag 0
HEN     EQU     $6              ; Host Enable

M_HRX   EQU     $FFFF94         ; SHI Receive FIFO
M_HCSR  EQU     $FFFF91         ; SHI Control/Status Register
M_HCKR  EQU     $FFFF90         ; SHI Clock Control Register
HRNE    EQU     17              ; SHI FIFO Not Empty flag
HI2C    EQU     1               ; SHI I2C Enable Control Bit
HCKFR   EQU     4               ; SHI I2C Clock Freeze Control Bit
HFM0    EQU     12              ; SHI I2C Filter Mode Bit 0
HFM1    EQU     13              ; SHI I2C Filter Mode Bit 1


        ORG PL:$ff0000,PL:$ff0000    ; bootstrap code starts at $ff0000

START
        movep #$0,X:M_OGDB      ; enable OnCE
        nop                     ; 5 NOP instructions, needed for test procedure
        nop
        nop
        nop
        nop
        clr a #$0,r5            ; clear a and init R5 with 0
        jset #MD,omr,OMR1XXX    ; If MD:MC:MB:MA=1xxx go to OMR1XXX
        jset #MC,omr,SHILD      ; If MD:MC:MB:MA=01xx, go load from SHI
        jclr #MB,omr,EPROMLD    ; If MD:MC:MB:MA=0001, go load from EPROM
        jset #MA,omr,RESERVED   ; If MD:MC:MB:MA=0011, go to RESERVED

;=========================================================================
; This is the routine that jumps to the internal Program ROM.
; MD:MC:MB:MA=0010

        move #PROMADDR,r1       ; store starting PROM address in r1

        bra     <FINISH

;=========================================================================
; This is the routine that loads from SHI.
; MD:MC:MB:MA=0100 - reserved for SHI
; MD:MC:MB:MA=0101 - Bootstrap from SHI (SPI slave)
; MD:MC:MB:MA=0110 - Bootstrap from SHI (I2C slave, HCKFR=1,100ns filter)
; MD:MC:MB:MA=0111 - Bootstrap from SHI (I2C slave, HCKFR=0)
```

```
SHILD

; This is the routine which loads a program through the SHI port.
; The SHI operates in the slave
; mode, with the 10-word FIFO enabled, and with the HREQ pin enabled for
; receive operation. The word size for transfer is 24 bits. The SHI
; operates in the SPI or in the I2C mode, according to the bootstrap mode.
;
; The program is downloaded according to the following rules:
; 1) 3 bytes - Define the program length.
; 2) 3 bytes - Define the address to which to start loading the program to.
; 3) 3n bytes (while n is the program length defined by the first 3 bytes)
; The program words will be stored in contiguous PRAM memory locations starting
; at the specified starting address.
; After storing the program words, program execution starts from the same
; address where loading started.

               move   #$A9,r1          ; prepare SHI control value in r1

; HEN=1, HI2C=0, HM1-HM0=10, HCKFR=0, HFIFO=1, HMST=0,
; HRQE1-HRQE0=01, HIDLE=0, HBIE=0, HTIE=0, HRIE1-HRIE0=00

               jclr   #MA,omr,SHI_CF     ; If MD:MC:MB:MA=01x0, go to SHI clock freeze

               jclr   #MB,omr,shi_loop   ; If MD:MC:MB:MA=0101, select SPI mode

               bset   #HI2C,r1           ; otherwise select I2C mode.

shi_loop       movep  r1,x:M_HCSR        ; enable SHI

               jclr   #HRNE,x:M_HCSR,*   ; wait for no. of words
               movep  x:M_HRX,a0

               jclr   #HRNE,x:M_HCSR,*   ; wait for starting address
               movep  x:M_HRX,r0
               move   r0,r1

               do     a0,_LOOP2
               jclr   #HRNE,x:M_HCSR,*   ; wait for HRX not empty
               movep  x:M_HRX,p:(r0)+    ; store in Program RAM
               nop                       ; req. because of restriction
_LOOP2

               bra    <FINISH

SHI_CF
               bset   #HI2C,r1           ; select I2C mode.
               bset   #HCKFR,r1          ; enable clock freeze in I2C mode.
               bset   #HFM0,x:M_HCKR     ; enable 100ns noise filter
               bset   #HFM1,x:M_HCKR     ; enable 100ns noise filter
               jset   #MB,omr,shi_loop   ; If MD:MC:MB:MA=0110, go to I2C load
               bra    <RESERVED          ; If MD:MC:MB:MA=0100, go to reserved

;=========================================================================
; This is the routine that loads from external EPROM.
```

```
; MD:MC:MB:MA=0001

EPROMLD

        move #BOOT,r2           ; r2 = address of external EPROM
        movep #AARV,X:M_AAR1    ; aar1 configured for SRAM types of access
        do #6,_LOOP9            ; read number of words and starting address
        movem p:(r2)+,a2        ; Get the 8 LSB from ext. P mem.
        asr #8,a,a              ; Shift 8 bit data into A1
_LOOP9                          ;
        move a1,r0              ; starting address for load
        move a1,r1              ; save it in r1
                                ; a0 holds the number of words
        do a0,_LOOP10           ; read program words
        do #3,_LOOP11           ; Each instruction has 3 bytes
        movem p:(r2)+,a2        ; Get the 8 LSB from ext. P mem.
        asr #8,a,a              ; Shift 8 bit data into A1
_LOOP11                         ; Go get another byte.
        movem a1,p:(r0)+        ; Store 24-bit result in P mem.
        nop                     ; pipeline delay
_LOOP10                         ; and go get another 24-bit word.
                                ; Boot from EPROM done

        bra     <FINISH



OMR1XXX
        jclr #MC,omr,BURN_RESER  ; IF MD:MC:MB:MA=101x, go to RESERVED
                                 ; IF MD:MC:MB:MA=1001, go to BURN
        jclr #MB,omr,OMR1IS0     ; IF MD:MC:MB:MA=110x, go to look for ISA/HC11
        jclr #MA,omr,I8051HOSTLD ; If MD:MC:MB:MA=1110, go load from 8051 Host
                                 ; If MD:MC:MB:MA=1111, go load from MC68302 Host


;=====================================================================
; This is the routine which loads a program through the HDI08 host port
; The program is downloaded from the host MCU with the following rules:
; 1) 3 bytes - Define the program length.
; 2) 3 bytes - Define the address to which to start loading the program to.
; 3) 3n bytes (while n is the program length defined by the 3 first bytes)
; The program words will be stored in contiguous PRAM memory locations starting
; at the specified starting address.
; After reading the program words, program execution starts from the same
; address where loading started.
; The host MCU may terminate the loading process by setting the HF1=0 and HF0=1.
; When the downloading is terminated, the program will start execution of the
; loaded program from the specified starting address.
; The HDI08 boot ROM program enables the following busses to download programs
; through the HDI08 port:
;
;  C - ISA           - Dual strobes non-multiplexed bus with negative strobe
;                    pulses dual positive request
;  D - HC11          - Single strobe non-multiplexed bus with positive strobe
;                    pulse single negative request.
```

```
;  E - i8051         - Dual strobes multiplexed bus with negative strobe pulses
;                    dual negative request.
;  F - MC68302       - Single strobe non-multiplexed bus with negative strobe
;                    pulse single negative request.
;=========================================================================

MC68302HOSTLD

        movep   #%0000000000111000,x:M_HPCR

                                ; Configure the following conditions:
                                ; HAP   = 0 Negative host acknowledge
                                ; HRP   = 0 Negative host request
                                ; HCSP  = 0 Negatice chip select input
                                ; HDDS  = 0 Single strobe bus (R/W~ and DS)
                                ; HMUX  = 0 Non multiplexed bus
                                ; HASP  = 0 (address strobe polarity has no
                                ;            meaning in non-multiplexed bus)
                                ; HDSP  = 0 Negative data stobes polarity
                                ; HROD  = 0 Host request is active when enabled
                                ; spare = 0 This bit should be set to 0 for
                                ;            future compatability
                                ; HEN   = 0 When the HPCR register is modified
                                ;            HEN should be cleared
                                ; HAEN  = 1 Host acknowledge is enabled
                                ; HREN  = 1 Host requests are enabled
                                ; HCSEN = 1 Host chip select input enabled
                                ; HA9EN = 0 (address 9 enable bit has no
                                ;            meaning in non-multiplexed bus)
                                ; HA8EN = 0 (address 8 enable bit has no
                                ;            meaning in non-multiplexed bus)
                                ; HGEN  = 0 Host GPIO pins are disabled

        bra     <HDI08CONT

OMR1IS0

        jset #MA,omr,HC11HOSTLD ; If MD:MC:MB:MA=1101, go load from HC11 Host
                                ; If MD:MC:MB:MA=1100, go load from ISA HOST

ISAHOSTLD

        movep   #%0101000000011000,x:M_HPCR

                                ; Configure the following conditions:
                                ; HAP   = 0 Negative host acknowledge
                                ; HRP   = 1 Positive host request
                                ; HCSP  = 0 Negatice chip select input
                                ; HDDS  = 1 Dual strobes bus (RD and WR)
                                ; HMUX  = 0 Non multiplexed bus
                                ; HASP  = 0 (address strobe polarity has no
                                ;            meaning in non-multiplexed bus)
                                ; HDSP  = 0 Negative data stobes polarity
                                ; HROD  = 0 Host request is active when enabled
                                ; spare = 0 This bit should be set to 0 for
```

```
                                ;           future compatability
                                ; HEN   = 0 When the HPCR register is modified
                                ;           HEN should be cleared
                                ; HAEN  = 0 Host acknowledge is disabled
                                ; HREN  = 1 Host requests are enabled
                                ; HCSEN = 1 Host chip select input enabled
                                ; HA9EN = 0 (address 9 enable bit has no
                                ;           meaning in non-multiplexed bus)
                                ; HA8EN = 0 (address 8 enable bit has no
                                ;           meaning non-multiplexed bus)
                                ; HGEN  = 0 Host GPIO pins are disabled

        bra     <HDI08CONT

HC11HOSTLD

        movep   #%0000001000011000,x:M_HPCR

                                ; Configure the following conditions:
                                ; HAP   = 0 Negative host acknowledge
                                ; HRP   = 0 Negative host request
                                ; HCSP  = 0 Negatice chip select input
                                ; HDDS  = 0 Single strobe bus (R/W~ and DS)
                                ; HMUX  = 0 Non multiplexed bus
                                ; HASP  = 0 (address strobe polarity has no
                                ;           meaning in non-multiplexed bus)
                                ; HDSP  = 1 Negative data stobes polarity
                                ; HROD  = 0 Host request is active when enabled
                                ; spare = 0 This bit should be set to 0 for
                                ;           future compatability
                                ; HEN   = 0 When the HPCR register is modified
                                ;           HEN should be cleared
                                ; HAEN  = 0 Host acknowledge is disabled
                                ; HREN  = 1 Host requests are enabled
                                ; HCSEN = 1 Host chip select input enabled
                                ; HA9EN = 0 (address 9 enable bit has no
                                ;           meaning in non-multiplexed bus)
                                ; HA8EN = 0 (address 8 enable bit has no
                                ;           meaning in non-multiplexed bus)
                                ; HGEN  = 0 Host GPIO pins are disabled

        bra     <HDI08CONT

I8051HOSTLD

        movep   #%0001110000011110,x:M_HPCR

                                ; Configure the following conditions:
                                ; HAP   = 0 Negative host acknowledge
                                ; HRP   = 0 Negatice host request
                                ; HCSP  = 0 Negatice chip select input
                                ; HDDS  = 1 Dual strobes bus (RD and WR)
                                ; HMUX  = 1 Multiplexed bus
                                ; HASP  = 1 Positive address strobe polarity
                                ; HDSP  = 0 Negative data stobes polarity
```

```
                                ; HROD  = 0 Host request is active when enabled
                                ; spare = 0 This bit should be set to 0 for
                                ;           future compatability
                                ; HEN   = 0 When the HPCR register is modified
                                ;           HEN should be cleared
                                ; HAEN  = 0 Host acknowledge is disabled
                                ; HREN  = 1 Host requests are enabled
                                ; HCSEN = 1 Host chip select input enabled
                                ; HA9EN = 1 Enable address 9 input
                                ; HA8EN = 1 Enable address 8 input
                                ; HGEN  = 0 Host GPIO pins are disabled

HDI08CONT

        bset    #HEN,x:M_HPCR           ; Enable the HDI08 to operate as host
                                        ; interface (set HEN=1)
        jclr    #HRDF,x:M_HSR,*         ; wait for the program length to be
                                        ; written

        movep   x:M_HORX,a0

        jclr    #HRDF,x:M_HSR,*         ; wait for the program starting address
                                        ; to be written

        movep   x:M_HORX,r0
        move    r0,r1

        do      a0,HDI08LOOP             ; set a loop with the downloaded length

HDI08LL

        jset    #HRDF,x:M_HSR,HDI08NW    ; If new word was loaded then jump to
                                        ; read that word
        jclr    #HF0,x:M_HSR,HDI08LL     ; If HF0=0 then continue with the
                                        ; downloading
        enddo                           ; Must terminate the do loop

        bra     <HDI08LOOP

HDI08NW

        movep   x:M_HORX,p:(r0)+        ; Move the new word into its destination
                                        ; location in the program RAM
        nop                             ; pipeline delay

HDI08LOOP


;=========================================================================
; This is the exit handler that returns execution to normal
; expanded mode and jumps to the RESET vector.

FINISH
        andi #$0,ccr            ; Clear CCR as if RESET to 0.
        jmp (r1)                ; Then go to starting Prog addr.
```

```
;=========================================================================
; MD:MC:MB:MA=1001 is used for Burn-in code

BURN_RESER

        jclr #MB,omr,BURN      ; IF MD:MC:MB:MA=1001, go to BURN

;=========================================================================
; The following modes are reserved, some of which are used for internal testing
; MD:MC:MB:MA=0011 is reserved
; MD:MC:MB:MA=1010 is reserved
; MD:MC:MB:MA=1011 is reserved

RESERVED

       bra     <*

;=========================================================================
; Code for burn-in
;=========================================================================

M_PCRC  EQU     $FFFFBF         ;; Port C GPIO Control Register
M_PDRC  EQU     $FFFFBD         ;; Port C GPIO Data Register
M_PRRC  EQU     $FFFFBE         ;; Port C Direction Register
SCKT    EQU     $3              ;; SCKT is GPIO bit #3 in ESAI (Port C)


EQUALDATA       equu     0      ;; 1 if xram and yram are of equal
                                ;; size and addresses, 0 otherwise.

        if      (EQUALDATA)
start_dram      equu     0      ;;
length_dram     equu     $1600  ;; same addresses
        else
start_xram      equu     0      ;; 13k XRAM
length_xram     equu     $3400
start_yram      equu     0      ;; 7k YRAM
length_yram     equu     $1c00
        endif

start_pram      equu     0      ;; 3k PRAM
length_pram     equu     $C00


BURN
                ;; get PATTERN pointer
                clr b   #PATTERNS,r6            ;; b is the error accumulator
                move    #<(NUM_PATTERNS-1),m6   ;; program runs forever in
                                                ;; cyclic form

                ;; configure SCKT as gpio output.
                movep   b,x:M_PDRC              ;; clear GPIO data register
                bclr    #SCKT,x:M_PCRC           ;; Define SCKT as output GPIO pin
                bset    #SCKT,x:M_PRRC           ;; SCKT toggles means test pass
```

```
                                        ;; r5 = test fail flag = $000000
                lua     (r5)-,r7        ;; r7 = test pass flag = $FFFFFF
burnin_loop
                     do #9,burn1
                ;;---------------------------
                ;; test RAM
                ;;  each pass checks 1 pattern
                ;;---------------------------
                move    p:(r6)+,x1      ;; pattern for x memory
                move    p:(r6)+,x0      ;; pattern for y memory
                move    p:(r6)+,y0      ;; pattern for p memory

                ;; write pattern to all memory locations

        if      (EQUALDATA)             ;; x/y ram symmetrical
                ;; write x and y memory
                clr a   #start_dram,r0  ;; start of x/y ram
                move    #>length_dram,n0 ;; length of x/y ram
                rep     n0
                mac x0,x1,a x,l:(r0)+   ;; exercise mac, write x/y ram

        else                            ;; x/y ram not symmetrical

                ;; write x memory
                clr a   #start_xram,r0  ;; start of xram
                move    #>length_xram,n0 ;; length of xram
                rep     n0
                mac x0,y0,a x1,x:(r0)+  ;; exercise mac, write xram

                ;; write y memory
                clr a   #start_yram,r1  ;; start of yram
                move    #>length_yram,n1 ;; length of yram
                rep     n1
                mac x1,y0,a x0,y:(r1)+  ;; exercise mac, write yram

        endif

                ;; write p memory
                clr a   #start_pram,r2  ;; start of pram
                move    #>length_pram,n2 ;; length of pram
                rep     n2
                move    y0,p:(r2)+      ;; write pram

                ;; check memory contents

        if      (EQUALDATA)             ;; x/y ram symmetrical

                ;; check dram
                clr a   #start_dram,r0  ;; restore pointer, clear a
                do      n0,_loopd
                move    x:(r0),a1       ;; a0=a2=0
                eor     x1,a
                add     a,b             ;; accumulate error in b
```

```
                move    y:(r0)+,a1              ;; a0=a2=0
                eor     x0,a
                add     a,b                     ;; accumulate error in b
_loopd

        else                                    ;; x/y ram not symmetrical

                ;; check xram
                clr a   #start_xram,r0          ;; restore pointer, clear a
                do      n0,_loopx
                move    x:(r0)+,a1              ;; a0=a2=0
                eor     x1,a
                add     a,b                     ;; accumulate error in b
_loopx

                ;; check yram
                clr a   #start_yram,r1          ;; restore pointer, clear a
                do      n1,_loopy
                move    y:(r1)+,a1              ;; a0=a2=0
                eor     x0,a
                add     a,b                     ;; accumulate error in b
_loopy

        endif

                ;; check pram
                clr a   #start_pram,r2          ;; restore pointer, clear a
                do      n2,_loopp
                move    p:(r2)+,a1              ;; a0=a2=0
                eor     y0,a
                add     a,b                     ;; accumulate error in b
_loopp
                ;;-----------------------------------------------------
                ;; toggle pin if no errors, stop execution otherwise.
                ;;-----------------------------------------------------
                ;; if error
                tne     r5,r7                   ;; r7=$FFFFFF as long as test pass
                                                ;; condition codes preserved
                                                ;; this instr can be removed in case of shortage
                movep   r7,x:M_OGDB             ;; write pass/fail flag to OnCE
                                                ;; condition codes preserved
                                                ;; this instr can be removed in case of shortage
                beq     label1
                bclr    #SCKT,x:M_PDRC          ;; clear SCKT if error,
                enddo                           ;; terminate the loop normally
                                                ;; this instr can be removed in case of shortage
                bra     <burn1                  ;; and stop execution
label1                                          ;; if no error
                bchg    #SCKT,x:M_PDRC          ;; toggle pin and keep on looping

burn1                                           ;; test completion
                debug                           ;; enter debug mode if OnCE port enabled
                                                ;; this instr can be removed in case of shortage
                wait                            ;; enter wait otherwise (OnCE port disabled)
```

```
BURN_END


                ORG PL:,PL:
PATTERNS        dsm     4                       ;; align for correct modulo addressing

               ORG  PL:BURN_END,PL:BURN_END

               dup PATTERNS-*                   ; write address in unused Boot ROM location
               dc *
               endm



                ORG     PL:PATTERNS,PL:PATTERNS ;; Each value is written to all memories

                dc      $555555
                dc      $AAAAAA
                dc      $333333
                dc      $F0F0F0

NUM_PATTERNS    equ     *-PATTERNS



;=====================================================================
; This code fills the unused bootstrap rom locations with their address

               dup $FF00C0-*
               dc *
               endm



;=====================================================================
; Reserved Area in the Program ROM: upper 128 words.
; Address range: $FFAF80 - $FFAFFF
;=====================================================================


               ORG PL:$FFAF80,PL:$FFAF80

; This code fills the unused rom locations with their address

               dup $FFB000-$14-*
               dc *
               endm

; Code segment for testing of ROM Patch
; This code segment is located in the uppermost addresses of the Program ROM

               ORG PL:$FFB000-$14,PL:$FFB000-$14

               move    #$80000,r0
               move    #$0,x0
```

```
        move    x0,x:(r0)+
        move    #$1,x0
        move    x0,x:(r0)+
        move    #$2,x0
        move    x0,x:(r0)+
        move    #$3,x0
        move    x0,x:(r0)+
        move    #$4,x0
        move    x0,x:(r0)+
        move    #$5,x0
        move    x0,x:(r0)+
        move    #$6,x0
        move    x0,x:(r0)+
        move    #$7,x0
        move    x0,x:(r0)+
        move    #$8,x0
        move    x0,x:(r0)+

        end
```

Appendix B Equates

```
;*****************************************************************************

;     EQUATES for DSP56366 interrupts

;     Last update: April 24, 2000

;

;*****************************************************************************

    page    132,55,0,0,0

    opt    mex

intequ  ident   1,0

    if    @DEF(I_VEC)

    ;leave user definition as is.

    else

I_VEC    equ    $0

    endif

;---------------------------------------------------------------------

; Non-Maskable interrupts

;---------------------------------------------------------------------

I_RESET  EQU  I_VEC+$00   ; Hardware RESET

I_STACK  EQU  I_VEC+$02   ; Stack Error

I_ILL    EQU  I_VEC+$04   ; Illegal Instruction

I_IINST    EQU  I_VEC+$04   ; Illegal Instruction

I_DBG    EQU  I_VEC+$06   ; Debug Request

I_TRAP   EQU  I_VEC+$08   ; Trap

I_NMI    EQU  I_VEC+$0A   ; Non Maskable Interrupt
```

```
;------------------------------------------------------------------------

; Interrupt Request Pins

;------------------------------------------------------------------------

I_IRQA   EQU  I_VEC+$10   ; IRQA

I_IRQB   EQU  I_VEC+$12   ; IRQB

I_IRQC   EQU  I_VEC+$14   ; IRQC

I_IRQD   EQU  I_VEC+$16   ; IRQD

;------------------------------------------------------------------------

; DMA Interrupts

;------------------------------------------------------------------------

I_DMA0   EQU  I_VEC+$18   ; DMA Channel 0

I_DMA1   EQU  I_VEC+$1A   ; DMA Channel 1

I_DMA2   EQU  I_VEC+$1C   ; DMA Channel 2

I_DMA3   EQU  I_VEC+$1E   ; DMA Channel 3

I_DMA4   EQU  I_VEC+$20   ; DMA Channel 4

I_DMA5   EQU  I_VEC+$22   ; DMA Channel 5

;------------------------------------------------------------------------

; DAX Interrupts

;------------------------------------------------------------------------

I_DAXTUE    EQU  I_VEC+$28   ; DAX Underrun Error

I_DAXBLK    EQU  I_VEC+$2A   ; DAX Block Transferred

I_DAXTD   EQU  I_VEC+$2E   ; DAX Audio Data Empty

;------------------------------------------------------------------------

; ESAI Interrupts

;------------------------------------------------------------------------

I_ESAIRD   EQU  I_VEC+$30   ; ESAI Receive Data

I_ESAIRED  EQU  I_VEC+$32   ; ESAI Receive Even Data

I_ESAIRDE  EQU  I_VEC+$34   ; ESAI Receive Data With Exception Status
```

```
I_ESAIRLS  EQU  I_VEC+$36   ; ESAI Receive Last Slot

I_ESAITD   EQU  I_VEC+$38   ; ESAI Transmit Data

I_ESAITED  EQU  I_VEC+$3A   ; ESAI Transmit Even Data

I_ESAITDE  EQU  I_VEC+$3C   ; ESAI Transmit Data With Exception Status

I_ESAITLS  EQU  I_VEC+$3E   ; ESAI Transmit Last Slot

;------------------------------------------------------------------------

; SHI Interrupts

;------------------------------------------------------------------------

I_SHITD    EQU  I_VEC+$40   ; SHI Transmit Data

I_SHITUE   EQU  I_VEC+$42   ; SHI Transmit Underrun Error

I_SHIRNE  EQU  I_VEC+$44   ; SHI Receive FIFO Not Empty

I_SHIRFF   EQU  I_VEC+$48   ; SHI Receive FIFO Full

I_SHIROE   EQU  I_VEC+$4A   ; SHI Receive Overrun Error

I_SHIBER    EQU  I_VEC+$4C   ; SHI Bus Error

;------------------------------------------------------------------------

; Timer Interrupts

;------------------------------------------------------------------------

I_TIM0C  EQU  I_VEC+$54   ; TIMER 0 compare

I_TIM0OF EQU  I_VEC+$56   ; TIMER 0 overflow

I_TIM1C  EQU  I_VEC+$58   ; TIMER 1 compare

I_TIM1OF EQU  I_VEC+$5A   ; TIMER 1 overflow

I_TIM2C  EQU  I_VEC+$5C   ; TIMER 2 compare

I_TIM2OF EQU  I_VEC+$5E   ; TIMER 2 overflow

;------------------------------------------------------------------------

; HDI08 Interrupts

;------------------------------------------------------------------------

I_HI08RX EQU  I_VEC+$60   ; Host Receive Data Full
```

```
I_HI08TX EQU  I_VEC+$62   ; Host Transmit Data Empty

I_HI08CM EQU  I_VEC+$64   ; Host Command (Default)

;------------------------------------------------------------------------

; ESAI_1 Interrupts

;------------------------------------------------------------------------

I_ESAI1RD   EQU  I_VEC+$70   ; ESAI_1 Receive Data

I_ESAI1RED  EQU  I_VEC+$72   ; ESAI_1 Receive Even Data

I_ESAI1RDE  EQU  I_VEC+$74   ; ESAI_1 Receive Data With Exception Status

I_ESAI1RLS  EQU  I_VEC+$76   ; ESAI_1 Receive Last Slot

I_ESAI1TD   EQU  I_VEC+$78   ; ESAI_1 Transmit Data

I_ESAI1TED  EQU  I_VEC+$7A   ; ESAI_1 Transmit Even Data

I_ESAI1TDE  EQU  I_VEC+$7C   ; ESAI_1 Transmit Data With Exception Status

I_ESAI1TLS  EQU  I_VEC+$7E   ; ESAI_1 Transmit Last Slot

;------------------------------------------------------------------------

; INTERRUPT ENDING ADDRESS

;------------------------------------------------------------------------

I_INTEND EQU  I_VEC+$FF   ; last address of interrupt vector space

;------------------ end of intequ.asm ------------------------

;*************************************************************************


;     EQUATES for DSP56366 I/O registers and ports

;     Last update: August 23, 2000

;

;*************************************************************************

    page    132,55,0,0,0

    opt    mex

ioequ   ident   1,0

;------------------------------------------------------------------------
```

```
;
;       EQUATES for I/O Port Programming
;
;---------------------------------------------------------------------
;       Register Addresses
M_HDR    EQU     $FFFFC9         ; Host port GPIO data Register
M_HDDR   EQU     $FFFFC8         ; Host port GPIO direction Register
M_PCRC   EQU     $FFFFBF         ; Port C Control Register
M_PRRC   EQU     $FFFFBE         ; Port C Direction Register
M_PDRC   EQU     $FFFFBD         ; Port C GPIO Data Register
M_PCRD   EQU     $FFFFD7         ; Port D Control register
M_PRRD   EQU     $FFFFD6         ; Port D Direction Data Register
M_PDRD   EQU     $FFFFD5         ; Port D GPIO Data Register
M_PCRE   EQU     $FFFF9F       ; Port E Control register
M_PRRE   EQU     $FFFF9E       ; Port E Direction Data Register
M_PDRE   EQU     $FFFF9D       ; Port E GPIO Data Register
M_OGDB   EQU     $FFFFFC         ; OnCE GDB Register
;---------------------------------------------------------------------
;
;       EQUATES for Exception Processing
;
;---------------------------------------------------------------------
;       Register Addresses
M_IPRC   EQU     $FFFFFF         ; Interrupt Priority Register Core
M_IPRP   EQU     $FFFFFE         ; Interrupt Priority Register Peripheral
;       Interrupt Priority Register Core (IPRC)
M_IAL    EQU     $7              ; IRQA Mode Mask
M_IAL0   EQU     0               ; IRQA Mode Interrupt Priority Level (low)
```

```
M_IAL1   EQU     1               ; IRQA Mode Interrupt Priority Level (high)
M_IAL2   EQU     2               ; IRQA Mode Trigger Mode
M_IBL    EQU     $38             ; IRQB Mode Mask
M_IBL0   EQU     3               ; IRQB Mode Interrupt Priority Level (low)
M_IBL1   EQU     4               ; IRQB Mode Interrupt Priority Level (high)
M_IBL2   EQU     5               ; IRQB Mode Trigger Mode
M_ICL    EQU     $1C0            ; IRQC Mode Mask
M_ICL0   EQU     6               ; IRQC Mode Interrupt Priority Level (low)
M_ICL1   EQU     7               ; IRQC Mode Interrupt Priority Level (high)
M_ICL2   EQU     8               ; IRQC Mode Trigger Mode
M_IDL    EQU     $E00            ; IRQD Mode Mask
M_IDL0   EQU     9               ; IRQD Mode Interrupt Priority Level (low)
M_IDL1   EQU     10              ; IRQD Mode Interrupt Priority Level (high)
M_IDL2   EQU     11              ; IRQD Mode Trigger Mode
M_D0L    EQU     $3000           ; DMA0 Interrupt priority Level Mask
M_D0L0   EQU     12              ; DMA0 Interrupt Priority Level (low)
M_D0L1   EQU     13              ; DMA0 Interrupt Priority Level (high)
M_D1L    EQU     $C000           ; DMA1 Interrupt Priority Level Mask
M_D1L0   EQU     14              ; DMA1 Interrupt Priority Level (low)
M_D1L1   EQU     15              ; DMA1 Interrupt Priority Level (high)
M_D2L    EQU     $30000          ; DMA2 Interrupt priority Level Mask
M_D2L0   EQU     16              ; DMA2 Interrupt Priority Level (low)
M_D2L1   EQU     17              ; DMA2 Interrupt Priority Level (high)
M_D3L    EQU     $C0000          ; DMA3 Interrupt Priority Level Mask
M_D3L0   EQU     18              ; DMA3 Interrupt Priority Level (low)
M_D3L1   EQU     19              ; DMA3 Interrupt Priority Level (high)
M_D4L    EQU     $300000         ; DMA4 Interrupt priority Level Mask
```

```
M_D4L0   EQU     20              ; DMA4 Interrupt Priority Level (low)
M_D4L1   EQU     21              ; DMA4 Interrupt Priority Level (high)
M_D5L    EQU     $C00000         ; DMA5 Interrupt priority Level Mask
M_D5L0   EQU     22              ; DMA5 Interrupt Priority Level (low)
M_D5L1   EQU     23              ; DMA5 Interrupt Priority Level (high)
;       Interrupt Priority Register Peripheral (IPRP)
M_ESL    EQU     $3              ; ESAI Interrupt Priority Level Mask
M_ESL0   EQU     0               ; ESAI Interrupt Priority Level (low)
M_ESL1   EQU     1               ; ESAI Interrupt Priority Level (high)
M_SHL    EQU     $C              ; SHI Interrupt Priority Level Mask
M_SHL0   EQU     2               ; SHI Interrupt Priority Level (low)
M_SHL1   EQU     3               ; SHI Interrupt Priority Level (high)
M_HDL    EQU     $30             ; HDI08 Interrupt Priority Level Mask
M_HDL0   EQU     4               ; HDI08 Interrupt Priority Level (low)
M_HDL1   EQU     5               ; HDI08 Interrupt Priority Level (high)
M_DAL    EQU     $C0             ; DAX Interrupt Priority Level  Mask
M_DAL0   EQU     6               ; DAX Interrupt Priority Level  (low)
M_DAL1   EQU     7               ; DAX Interrupt Priority Level  (high)
M_TAL   EQU     $300            ;Timer Interrupt Priority Level Mask
M_TAL0   EQU     8                ;Timer Interrupt Priority Level (low)
M_TAL1   EQU     9                ;Timer Interrupt Priority Level (high)
M_ES1L    EQU     $C00          ; ESAI_1 Interrupt Priority Level Mask
M_ESL10   EQU     0               ; ESAI_1 Interrupt Priority Level (low)
M_ESL11   EQU     1               ; ESAI_1 Interrupt Priority Level (high)
;---------------------------------------------------------------------
;
;       EQUATES for Direct Memory Access (DMA)
;
```

```
;---------------------------------------------------------------------

;       Register Addresses Of DMA

M_DSTR   EQU     $FFFFF4         ; DMA Status Register

M_DOR0   EQU     $FFFFF3         ; DMA Offset Register 0

M_DOR1   EQU     $FFFFF2         ; DMA Offset Register 1

M_DOR2   EQU     $FFFFF1         ; DMA Offset Register 2

M_DOR3   EQU     $FFFFF0         ; DMA Offset Register 3

;       Register Addresses Of DMA0

M_DSR0   EQU     $FFFFEF         ; DMA0 Source Address Register

M_DDR0   EQU     $FFFFEE         ; DMA0 Destination Address Register

M_DCO0   EQU     $FFFFED         ; DMA0 Counter

M_DCR0   EQU     $FFFFEC         ; DMA0 Control Register

;       Register Addresses Of DMA1

M_DSR1   EQU     $FFFFEB         ; DMA1 Source Address Register

M_DDR1   EQU     $FFFFEA         ; DMA1 Destination Address Register

M_DCO1   EQU     $FFFFE9         ; DMA1 Counter

M_DCR1   EQU     $FFFFE8         ; DMA1 Control Register

;       Register Addresses Of DMA2

M_DSR2   EQU     $FFFFE7         ; DMA2 Source Address Register

M_DDR2   EQU     $FFFFE6         ; DMA2 Destination Address Register

M_DCO2   EQU     $FFFFE5         ; DMA2 Counter

M_DCR2   EQU     $FFFFE4         ; DMA2 Control Register


;       Register Addresses Of DMA3

M_DSR3   EQU     $FFFFE3         ; DMA3 Source Address Register

M_DDR3   EQU     $FFFFE2         ; DMA3 Destination Address Register

M_DCO3   EQU     $FFFFE1         ; DMA3 Counter
```

```
M_DCR3   EQU     $FFFFE0         ; DMA3 Control Register
;       Register Addresses Of DMA4
M_DSR4   EQU     $FFFFDF         ; DMA4 Source Address Register
M_DDR4   EQU     $FFFFDE         ; DMA4 Destination Address Register
M_DCO4   EQU     $FFFFDD         ; DMA4 Counter
M_DCR4   EQU     $FFFFDC         ; DMA4 Control Register
;       Register Addresses Of DMA5
M_DSR5   EQU     $FFFFDB         ; DMA5 Source Address Register
M_DDR5   EQU     $FFFFDA         ; DMA5 Destination Address Register
M_DCO5   EQU     $FFFFD9         ; DMA5 Counter
M_DCR5   EQU     $FFFFD8         ; DMA5 Control Register
;       DMA Control Register
M_DSS    EQU     $3              ; DMA Source Space Mask (DSS0-Dss1)
M_DSS0   EQU     0               ; DMA Source Memory space 0
M_DSS1   EQU     1               ; DMA Source Memory space 1
M_DDS    EQU     $C              ; DMA Destination Space Mask (DDS-DDS1)
M_DDS0   EQU     2               ; DMA Destination Memory Space 0
M_DDS1   EQU     3               ; DMA Destination Memory Space 1
M_DAM    EQU     $3f0            ; DMA Address Mode Mask (DAM5-DAM0)
M_DAM0   EQU     4               ; DMA Address Mode 0
M_DAM1   EQU     5               ; DMA Address Mode 1
M_DAM2   EQU     6               ; DMA Address Mode 2
M_DAM3   EQU     7               ; DMA Address Mode 3
M_DAM4   EQU     8               ; DMA Address Mode 4
M_DAM5   EQU     9               ; DMA Address Mode 5
M_D3D    EQU     10              ; DMA Three Dimensional Mode
M_DRS    EQU     $F800           ; DMA Request Source Mask (DRS0-DRS4)
M_DRS0    EQU     11                ;DMA Request Source bit 0
```

```
M_DRS1    EQU     12                ;DMA Request Source bit 1
M_DRS2    EQU     13                ;DMA Request Source bit 2
M_DRS3    EQU     14                ;DMA Request Source bit 3
M_DRS4    EQU     15                ;DMA Request Source bit 4
M_DCON    EQU     16              ; DMA Continuous Mode
M_DPR     EQU     $60000          ; DMA Channel Priority
M_DPR0    EQU     17              ; DMA Channel Priority Level (low)
M_DPR1    EQU     18              ; DMA Channel Priority Level (high)
M_DTM     EQU     $380000         ; DMA Transfer Mode Mask (DTM2-DTM0)
M_DTM0    EQU     19              ; DMA Transfer Mode 0
M_DTM1    EQU     20              ; DMA Transfer Mode 1
M_DTM2    EQU     21              ; DMA Transfer Mode 2
M_DIE     EQU     22              ; DMA Interrupt Enable bit
M_DE      EQU     23              ; DMA Channel Enable bit
;       DMA Status Register
M_DTD     EQU     $3F             ; Channel Transfer Done Status MASK (DTD0-DTD5)
M_DTD0    EQU     0               ; DMA Channel Transfer Done Status 0
M_DTD1    EQU     1               ; DMA Channel Transfer Done Status 1
M_DTD2    EQU     2               ; DMA Channel Transfer Done Status 2
M_DTD3    EQU     3               ; DMA Channel Transfer Done Status 3
M_DTD4    EQU     4               ; DMA Channel Transfer Done Status 4
M_DTD5    EQU     5               ; DMA Channel Transfer Done Status 5
M_DACT    EQU     8               ; DMA Active State
M_DCH     EQU     $E00            ; DMA Active Channel Mask (DCH0-DCH2)
M_DCH0    EQU     9               ; DMA Active Channel 0
M_DCH1    EQU     10              ; DMA Active Channel 1
M_DCH2    EQU     11              ; DMA Active Channel 2
```

```
;------------------------------------------------------------------------

;

;       EQUATES for Phase Locked Loop (PLL)

;

;------------------------------------------------------------------------

;       Register Addresses Of PLL

M_PCTL   EQU     $FFFFFD         ; PLL Control Register

;       PLL Control Register

M_MF     EQU     $FFF            ; Multiplication Factor Bits Mask (MF0-MF11)

M_MF0    EQU   0                     ;Multiplication Factor bit 0

M_MF1    EQU   1                     ;Multiplication Factor bit 1

M_MF2    EQU   2                     ;Multiplication Factor bit 2

M_MF3    EQU   3                     ;Multiplication Factor bit 3

M_MF4    EQU   4                     ;Multiplication Factor bit 4

M_MF5    EQU   5                     ;Multiplication Factor bit 5

M_MF6    EQU   6                     ;Multiplication Factor bit 6

M_MF7    EQU   7                     ;Multiplication Factor bit 7

M_MF8    EQU   8                     ;Multiplication Factor bit 8

M_MF9    EQU   9                     ;Multiplication Factor bit 9

M_MF10    EQU   10                     ;Multiplication Factor bit 10

M_MF11    EQU   11                     ;Multiplication Factor bit 11

M_DF     EQU     $7000           ; Division Factor Bits Mask (DF0-DF2)

M_DF0    EQU   12                     ;Division Factor bit 0

M_DF1    EQU   13                     ;Division Factor bit 1

M_DF2    EQU   14                     ;Division Factor bit 2

M_XTLR   EQU     15              ; XTAL Range select bit

M_XTLD   EQU     16              ; XTAL Disable Bit

M_PSTP   EQU     17              ; STOP Processing State Bit
```

```
M_PEN    EQU     18              ; PLL Enable Bit
M_COD   EQU     19              ; PLL Clock Output Disable Bit
M_PD     EQU     $F00000         ; PreDivider Factor Bits Mask (PD0-PD3)
M_PD0    EQU   20                    ;PreDivider Factor bit 0
M_PD1    EQU   21                    ;PreDivider Factor bit 1
M_PD2    EQU   22                    ;PreDivider Factor bit 2
M_PD3    EQU   23                    ;PreDivider Factor bit 3
;---------------------------------------------------------------------
;
;       EQUATES for BIU
;
;---------------------------------------------------------------------
;       Register Addresses Of BIU
M_BCR    EQU     $FFFFFB         ; Bus Control Register
M_DCR    EQU     $FFFFFA         ; DRAM Control Register
M_AAR0   EQU     $FFFFF9         ; Address Attribute Register 0
M_AAR1   EQU     $FFFFF8         ; Address Attribute Register 1
M_AAR2   EQU     $FFFFF7         ; Address Attribute Register 2
M_AAR3   EQU     $FFFFF6         ; Address Attribute Register 3
M_IDR    EQU     $FFFFF5         ; ID Register
;       Bus Control Register
M_BA0W   EQU     $1F             ; Area 0 Wait Control Mask (BA0W0-BA0W4)
M_BA0W0   EQU   0                ;Area 0 Wait Control Bit 0
M_BA0W1   EQU   1                ;Area 0 Wait Control Bit 1
M_BA0W2   EQU   2                ;Area 0 Wait Control Bit 2
M_BA0W3   EQU   3                ;Area 0 Wait Control Bit 3
M_BA0W4   EQU   4                ;Area 0 Wait Control Bit 4
```

```
M_BA1W   EQU     $3E0            ; Area 1 Wait Control Mask (BA1W0-BA14)
M_BA1W0   EQU   5                ;Area 1 Wait Control Bit 0
M_BA1W1   EQU   6               ;Area 1 Wait Control Bit 1
M_BA1W2   EQU   7                ;Area 1 Wait Control Bit 2
M_BA1W3   EQU   8                ;Area 1 Wait Control Bit 3
M_BA1W4   EQU   9                ;Area 1 Wait Control Bit 4
M_BA2W   EQU     $1C00           ; Area 2 Wait Control Mask (BA2W0-BA2W2)
M_BA2W0   EQU   10                ;Area 2 Wait Control Bit 0
M_BA2W1   EQU   11                ;Area 2 Wait Control Bit 1
M_BA2W2   EQU   12                ;Area 2 Wait Control Bit 2
M_BA3W   EQU     $E000           ; Area 3 Wait Control Mask (BA3W0-BA3W3)
M_BA3W0   EQU   13                ;Area 3 Wait Control Bit 0
M_BA3W1   EQU   14                ;Area 3 Wait Control Bit 1
M_BA3W2   EQU   15                ;Area 3 Wait Control Bit 2
M_BDFW   EQU     $1F0000         ; Default Area Wait Control Mask (BDFW0-BDFW4)
M_BDFW0   EQU     16               ;Default Area Wait Control bit 0
M_BDFW1   EQU     17               ;Default Area Wait Control bit 1
M_BDFW2   EQU     18               ;Default Area Wait Control bit 2
M_BDFW3   EQU     19               ;Default Area Wait Control bit 3
M_BDFW4   EQU     20               ;Default Area Wait Control bit 4
M_BBS    EQU     21              ; Bus State
M_BLH    EQU     22              ; Bus Lock Hold
M_BRH    EQU     23              ; Bus Request Hold
;       DRAM Control Register
M_BCW    EQU     $3              ; In Page Wait States Bits Mask (BCW0-BCW1)
M_BCW0    EQU     0              ; In Page Wait States Bit 0
M_BCW1   EQU     1              ; In Page Wait States Bit 1
M_BRW    EQU     $C              ; Out Of Page Wait States Bits Mask (BRW0-BRW1)
```

```
M_BRW0    EQU     2             ;Out of Page Wait States bit 0
M_BRW1    EQU     3             ; Out of Page Wait States bit 1
M_BPS     EQU     $300           ; DRAM Page Size Bits Mask (BPS0-BPS1)
M_BPS0    EQU     4            ; DRAM Page Size Bits 0
M_BPS1    EQU     5            ; DRAM Page Size Bits 1
M_BPLE    EQU     11             ; Page Logic Enable
M_BME     EQU     12             ; Mastership Enable
M_BRE     EQU     13             ; Refresh Enable
M_BSTR    EQU     14             ; Software Triggered Refresh
M_BRF     EQU     $7F8000         ; Refresh Rate Bits Mask (BRF0-BRF7)
M_BRF0    EQU     15         ; Refresh Rate Bit 0
M_BRF1    EQU     16         ; Refresh Rate Bit 1
M_BRF2    EQU     17         ; Refresh Rate Bit 2
M_BRF3    EQU     18         ; Refresh Rate Bit 3
M_BRF4    EQU     19         ; Refresh Rate Bit 4
M_BRF5    EQU     20         ; Refresh Rate Bit 5
M_BRF6    EQU     21         ; Refresh Rate Bit 6
M_BRF7    EQU     22         ; Refresh Rate Bit 7
M_BRP     EQU     23             ; Refresh prescaler
;       Address Attribute Registers
M_BAT     EQU     $3             ; External Access Type and Pin Definition Bits Mask (BAT0-BAT1)
M_BAT0    EQU     0              ; External Access Type and Pin Definition Bits 0
M_BAT1    EQU     1              ; External Access Type and Pin Definition Bits 1
M_BAAP    EQU     2              ; Address Attribute Pin Polarity
M_BPEN    EQU     3              ; Program Space Enable
M_BXEN    EQU     4              ; X Data Space Enable
M_BYEN    EQU     5              ; Y Data Space Enable
```

```
M_BAM    EQU     6               ; Address Muxing
M_BPAC   EQU     7               ; Packing Enable
M_BNC    EQU     $F00            ; Number of Address Bits to Compare Mask (BNC0-BNC3)
M_BNC0    EQU     8             ; Number of Address Bits to Compare 0
M_BNC1    EQU     9             ; Number of Address Bits to Compare 1
M_BNC2    EQU     10             ; Number of Address Bits to Compare 2
M_BNC3    EQU     11             ; Number of Address Bits to Compare 3
M_BAC    EQU     $FFF000         ; Address to Compare Bits Mask (BAC0-BAC11)
M_BAC0    EQU     12           ; Address to Compare Bits 0
M_BAC1    EQU     13           ; Address to Compare Bits 1
M_BAC2    EQU     14         ; Address to Compare Bits 2
M_BAC3    EQU     15           ; Address to Compare Bits 3
M_BAC4    EQU     16           ; Address to Compare Bits 4
M_BAC5    EQU     17           ; Address to Compare Bits 5
M_BAC6    EQU     18           ; Address to Compare Bits 6
M_BAC7    EQU     19           ; Address to Compare Bits 7
M_BAC8    EQU     20           ; Address to Compare Bits 8
M_BAC9    EQU     21           ; Address to Compare Bits 9
M_BAC10    EQU     22           ; Address to Compare Bits 10
M_BAC11    EQU     23           ; Address to Compare Bits 11
;       control and status bits in SR
M_C     EQU     0               ; Carry
M_V      EQU     1              ; Overflow
M_Z      EQU     2              ; Zero
M_N      EQU     3              ; Negative
M_U      EQU     4              ; Unnormalized
M_E      EQU     5              ; Extension
M_L      EQU     6              ; Limit
```

```
M_S      EQU     7               ; Scaling Bit
M_I0     EQU     8               ; Interupt Mask Bit 0
M_I1     EQU     9               ; Interupt Mask Bit 1
M_S0     EQU     10              ; Scaling Mode Bit 0
M_S1     EQU     11              ; Scaling Mode Bit 1
M_SC     EQU     13              ; Sixteen_Bit Compatibility
M_DM     EQU     14              ; Double Precision Multiply
M_LF     EQU     15              ; DO-Loop Flag
M_FV     EQU     16              ; DO-Forever Flag
M_SA     EQU     17              ; Sixteen-Bit Arithmetic
M_CE     EQU     19              ; Instruction Cache Enable
M_SM     EQU     20              ; Arithmetic Saturation
M_RM     EQU     21              ; Rounding Mode
M_CP     EQU     $c00000         ; mask for CORE-DMA priority bits in SR
M_CP0    EQU     22              ; bit 0 of priority bits in SR
M_CP1    EQU     23              ; bit 1 of priority bits in SR
;       control and status bits in OMR
M_MA     EQU     0               ; Operating Mode A
M_MB     EQU     1               ; Operating Mode B
M_MC     EQU     2               ; Operating Mode C
M_MD     EQU     3               ; Operating Mode D
M_EBD    EQU     4               ; External Bus Disable bit in OMR
M_SD     EQU     6               ; Stop Delay
M_MS     EQU     7                 ;Memory Switch Mode
M_CDP    EQU     $300            ; mask for CORE-DMA priority bits in OMR
M_CDP0   EQU     8               ; bit 0 of priority bits in OMR Core DMA
M_CDP1   EQU     9               ; bit 1 of priority bits in OMR Core DMA
```

```
M_BE    EQU     10              ; Burst Enable
M_TAS    EQU     11              ; TA Synchronize Select
M_BRT    EQU     12              ; Bus Release Timing
M_ABE   EQU     13              ;Async. Bus Arbitration Enable
M_APD   EQU     14              ;Addess Priority Disable
M_ATE   EQU     15              ;Address Tracing Enable
M_XYS    EQU     16              ; Stack Extension space select bit in OMR.
M_EUN    EQU     17              ; Extensed stack UNderflow flag in OMR.
M_EOV    EQU     18              ; Extended stack OVerflow flag in OMR.
M_WRP    EQU     19              ; Extended WRaP flag in OMR.
M_SEN    EQU     20              ; Stack Extension Enable bit in OMR.
M_PAEN   EQU      23             ; Patch Enable
;---------------------------------------------------------------------
;
;       EQUATES for DAX (SPDIF Tx)
;
;---------------------------------------------------------------------
;       Register Addresses


M_XSTR   EQU     $FFFFD4         ; DAX Status Register (XSTR)
M_XADRB  EQU     $FFFFD3         ; DAX Audio Data Register B (XADRB)
M_XADR    EQU     $FFFFD2          ;DAX Audio Data Register (XADR)
M_XADRA  EQU     $FFFFD2         ; DAX Audio Data Register A (XADRA)
M_XNADR  EQU     $FFFFD1         ; DAX Non-Audio Data Register (XNADR)
M_XCTR   EQU     $FFFFD0         ; DAX Control Register (XCTR)
;       status bits in XSTR
M_XADE   EQU     0               ; DAX Audio Data Register Empty (XADE)
M_XAUR   EQU     1               ; DAX Trasmit Underrun Error Flag (XAUR)
```

```
M_XBLK   EQU     2               ; DAX Block Transferred (XBLK)
;       non-audio bits in XNADR
M_XVA    EQU     10              ; DAX Channel A Validity (XVA)
M_XUA    EQU     11              ; DAX Channel A User Data (XUA)
M_XCA    EQU     12              ; DAX Channel A Channel Status (XCA)
M_XVB    EQU     13              ; DAX Channel B Validity (XVB)
M_XUB    EQU     14              ; DAX Channel B User Data (XUB)
M_XCB    EQU     15              ; DAX Channel B Channel Status (XCB)
;       control bits in XCTR
M_XDIE   EQU     0               ; DAX Audio Data Register Empty Interrupt Enable (XDIE)
M_XUIE   EQU     1               ; DAX Underrun Error Interrupt Enable (XUIE)
M_XBIE   EQU     2               ; DAX Block Transferred Interrupt Enable (XBIE)
M_XCS0   EQU     3               ; DAX Clock Input Select 0 (XCS0)
M_XCS1   EQU     4               ; DAX Clock Input Select 1 (XCS1)
M_XSB    EQU     5               ; DAX Start Block (XSB)
;------------------------------------------------------------------------
;
;       EQUATES for SHI
;
;------------------------------------------------------------------------
;       Register Addresses


M_HRX    EQU     $FFFF94         ; SHI Receive FIFO (HRX)
M_HTX    EQU     $FFFF93         ; SHI Transmit Register (HTX)
M_HSAR   EQU     $FFFF92         ; SHI I2C Slave Address Register (HSAR)
M_HCSR   EQU     $FFFF91         ; SHI Control/Status Register (HCSR)
M_HCKR   EQU     $FFFF90         ; SHI Clock Control Register (HCKR)
```

```
;       HSAR bits

M_HA6    EQU     23              ; SHI I2C Slave Address (HA6)

M_HA5    EQU     22              ; SHI I2C Slave Address (HA5)

M_HA4    EQU     21              ; SHI I2C Slave Address (HA4)

M_HA3    EQU     20              ; SHI I2C Slave Address (HA3)

M_HA1    EQU     18              ; SHI I2C Slave Address (HA1)

;      control and status bits in HCSR

M_HBUSY  EQU     22              ; SHI Host Busy (HBUSY)

M_HBER   EQU     21              ; SHI Bus Error (HBER)

M_HROE   EQU     20              ; SHI Receive Overrun Error (HROE)

M_HRFF   EQU     19              ; SHI Receivr FIFO Full (HRFF)

M_HRNE   EQU     17              ; SHI Receive FIFO Not Empty (HRNE)

M_HTDE   EQU     15              ; SHI Host Transmit data Empty (HTDE)

M_HTUE   EQU     14              ; SHI Host Transmit Underrun Error (HTUE)

M_HRIE1  EQU     13              ; SHI Receive Interrupt Enable (HRIE1)

M_HRIE0  EQU     12              ; SHI Receive Interrupt Enable (HRIE0)

M_HTIE   EQU     11              ; SHI Transmit Interrupt Enable (HTIE)

M_HBIE   EQU     10              ; SHI Bus-Error Interrupt Enable (HBIE)

M_HIDLE  EQU     9               ; SHI Idle (HIDLE)

M_HRQE1  EQU     8               ; SHI Host Request Enable (HRQE1)

M_HRQE0  EQU     7               ; SHI Host Request Enable (HRQE0)

M_HMST   EQU     6               ; SHI Master Mode (HMST)

M_HFIFO  EQU     5               ; SHI FIFO Enable Control (HFIFO)

M_HCKFR  EQU     4               ; SHI Clock Freeze (HCKFR)

M_HM1    EQU     3               ; SHI Serial Host Interface Mode (HM1)

M_HM0    EQU     2               ; SHI Serial Host Interface Mode (HM0)

M_HI2C   EQU     1               ; SHI I2c/SPI Selection (HI2C)

M_HEN    EQU     0               ; SHI Host Enable (HEN)
```

```
;       control bits in HCKR

M_HFM1   EQU     13              ; SHI Filter Model (HFM1)

M_HFM0   EQU     12              ; SHI Filter Model (HFM0)

M_HDM7   EQU     10               ; SHI Divider Modulus Select (HDM7)

M_HDM6   EQU     9               ; SHI Divider Modulus Select (HDM6)

M_HDM5   EQU     8               ; SHI Divider Modulus Select (HDM5)

M_HDM4   EQU     7               ; SHI Divider Modulus Select (HDM4)

M_HDM3   EQU     6               ; SHI Divider Modulus Select (HDM3)

M_HDM2   EQU     5               ; SHI Divider Modulus Select (HDM2)

M_HDM1   EQU     4               ; SHI Divider Modulus Select (HDM1)

M_HDM0   EQU     3               ; SHI Divider Modulus Select (HDM0)

M_HRS    EQU     2               ; SHI Prescalar Rate Select (HRS)

M_CPOL   EQU     1               ; SHI Clock Polarity (CPOL)

M_CPHA   EQU     0               ; SHI Clock Phase (CPHA)


;-------------------------------------------------------------------------

;

;       EQUATES for ESAI_1 Registers

; register bit equates can be the same as for the ESAI register bit equates.

;-------------------------------------------------------------------------

;       Register Addresses


M_EMUXR  EQU    $FFFFAF         ; MUX PIN CONTROL REGISTER (EMUXR)

M_RSMB_1 EQU     $FFFF9C         ; ESAI_1 Receive Slot Mask Register B (RSMB_1)

M_RSMA_1 EQU     $FFFF9B         ; ESAI_1 Receive Slot Mask Register A (RSMA_1)

M_TSMB_1 EQU     $FFFF9A         ; ESAI_1 Transmit Slot Mask Register B (TSMB_1)

M_TSMA_1 EQU     $FFFF99         ; ESAI_1 Transmit Slot Mask Register A (TSMA_1)

M_RCCR_1 EQU     $FFFF98         ; ESAI_1 Receive Clock Control Register (RCCR_1)
```

```
M_RCR_1  EQU     $FFFF97         ; ESAI_1 Receive Control Register (RCR_1)
M_TCCR_1 EQU     $FFFF96         ; ESAI_1 Transmit Clock Control Register (TCCR_1)
M_TCR_1  EQU     $FFFF95         ; ESAI_1 Transmit Control Register (TCR_1)
M_SAICR_1 EQU    $FFFF94         ; ESAI_1 Control Register (SAICR_1)
M_SAISR_1 EQU     $FFFF93         ; ESAI_1 Status Register (SAISR_1)
M_RX3_1 EQU     $FFFF8B         ; ESAI_1 Receive Data Register 3 (RX3_1)
M_RX2_1 EQU     $FFFF8A         ; ESAI_1 Receive Data Register 2 (RX2_1)
M_RX1_1 EQU     $FFFF89         ; ESAI_1 Receive Data Register 1 (RX1_1)
M_RX0_1 EQU     $FFFF88         ; ESAI_1 Receive Data Register 0 (RX0_1)
M_TSR_1 EQU     $FFFF86         ; ESAI_1 Time Slot Register (TSR_1)
M_TX5_1 EQU     $FFFF85         ; ESAI_1 Transmit Data Register 5 (TX5_1)
M_TX4_1 EQU     $FFFF84         ; ESAI_1 Transmit Data Register 4 (TX4_1)
M_TX3_1 EQU     $FFFF83         ; ESAI_1 Transmit Data Register 3 (TX3_1)
M_TX2_1 EQU     $FFFF82         ; ESAI_1 Transmit Data Register 2 (TX2_1)
M_TX1_1 EQU     $FFFF81         ; ESAI_1 Transmit Data Register 1 (TX1_1)
M_TX0_1 EQU     $FFFF80         ; ESAI_1 Transmit Data Register 0 (TX0_1)
;------------------------------------------------------------------------
;
;       EQUATES for ESAI
;
;------------------------------------------------------------------------
;       Register Addresses


M_RSMB   EQU     $FFFFBC         ; ESAI Receive Slot Mask Register B (RSMB)
M_RSMA   EQU     $FFFFBB         ; ESAI Receive Slot Mask Register A (RSMA)
M_TSMB   EQU     $FFFFBA         ; ESAI Transmit Slot Mask Register B (TSMB)
M_TSMA   EQU     $FFFFB9         ; ESAI Transmit Slot Mask Register A (TSMA)
M_RCCR   EQU     $FFFFB8         ; ESAI Receive Clock Control Register (RCCR)
```

```
M_RCR    EQU     $FFFFB7         ; ESAI Receive Control Register (RCR)
M_TCCR   EQU     $FFFFB6         ; ESAI Transmit Clock Control Register (TCCR)
M_TCR    EQU     $FFFFB5         ; ESAI Transmit Control Register (TCR)
M_SAICR  EQU     $FFFFB4         ; ESAI Control Register (SAICR)
M_SAISR  EQU     $FFFFB3         ; ESAI Status Register (SAISR)
M_RX3    EQU     $FFFFAB         ; ESAI Receive Data Register 3 (RX3)
M_RX2    EQU     $FFFFAA         ; ESAI Receive Data Register 2 (RX2)
M_RX1    EQU     $FFFFA9         ; ESAI Receive Data Register 1 (RX1)
M_RX0    EQU     $FFFFA8         ; ESAI Receive Data Register 0 (RX0)
M_TSR    EQU     $FFFFA6         ; ESAI Time Slot Register (TSR)
M_TX5    EQU     $FFFFA5         ; ESAI Transmit Data Register 5 (TX5)
M_TX4    EQU     $FFFFA4         ; ESAI Transmit Data Register 4 (TX4)
M_TX3    EQU     $FFFFA3         ; ESAI Transmit Data Register 3 (TX3)
M_TX2    EQU     $FFFFA2         ; ESAI Transmit Data Register 2 (TX2)
M_TX1    EQU     $FFFFA1         ; ESAI Transmit Data Register 1 (TX1)
M_TX0    EQU     $FFFFA0         ; ESAI Transmit Data Register 0 (TX0)
;       RSMB Register bits
M_RS31   EQU     15              ; ESAI
M_RS30   EQU     14              ; ESAI
M_RS29   EQU     13              ; ESAI
M_RS28   EQU     12              ; ESAI
M_RS27   EQU     11              ; ESAI
M_RS26   EQU     10              ; ESAI
M_RS25   EQU     9               ; ESAI
M_RS24   EQU     8               ; ESAI
M_RS23   EQU     7               ; ESAI
M_RS22   EQU     6               ; ESAI
```

```
M_RS21   EQU     5               ; ESAI

M_RS20   EQU     4               ; ESAI

M_RS19   EQU     3               ; ESAI

M_RS18   EQU     2               ; ESAI

M_RS17   EQU     1               ; ESAI

M_RS16   EQU     0               ; ESAI

;       RSMA Register bits

M_RS15   EQU     15              ; ESAI

M_RS14   EQU     14              ; ESAI

M_RS13   EQU     13              ; ESAI

M_RS12   EQU     12              ; ESAI

M_RS11   EQU     11              ; ESAI

M_RS10   EQU     10              ; ESAI

M_RS9    EQU     9               ; ESAI

M_RS8    EQU     8               ; ESAI

M_RS7    EQU     7               ; ESAI

M_RS6    EQU     6               ; ESAI

M_RS5    EQU     5               ; ESAI

M_RS4    EQU     4               ; ESAI

M_RS3    EQU     3               ; ESAI

M_RS2    EQU     2               ; ESAI

M_RS1    EQU     1               ; ESAI

M_RS0    EQU     0               ; ESAI

;       TSMB Register bits

M_TS31   EQU     15              ; ESAI

M_TS30   EQU     14              ; ESAI

M_TS29   EQU     13              ; ESAI

M_TS28   EQU     12              ; ESAI
```

```
M_TS27   EQU     11              ; ESAI

M_TS26   EQU     10              ; ESAI

M_TS25   EQU     9               ; ESAI

M_TS24   EQU     8               ; ESAI

M_TS23   EQU     7               ; ESAI

M_TS22   EQU     6               ; ESAI

M_TS21   EQU     5               ; ESAI

M_TS20   EQU     4               ; ESAI

M_TS19   EQU     3               ; ESAI

M_TS18   EQU     2               ; ESAI

M_TS17   EQU     1               ; ESAI

M_TS16   EQU     0               ; ESAI

;       TSMA Register bits

M_TS15   EQU     15              ; ESAI

M_TS14   EQU     14              ; ESAI

M_TS13   EQU     13              ; ESAI

M_TS12   EQU     12              ; ESAI

M_TS11   EQU     11              ; ESAI

M_TS10   EQU     10              ; ESAI

M_TS9    EQU     9               ; ESAI

M_TS8    EQU     8               ; ESAI

M_TS7    EQU     7               ; ESAI

M_TS6    EQU     6               ; ESAI

M_TS5    EQU     5               ; ESAI

M_TS4    EQU     4               ; ESAI

M_TS3    EQU     3               ; ESAI

M_TS2    EQU     2               ; ESAI
```

```
M_TS1    EQU     1               ; ESAI

M_TS0    EQU     0               ; ESAI

;       RCCR Register bits

M_RHCKD  EQU     23              ; ESAI

M_RFSD   EQU     22              ; ESAI

M_RCKD   EQU     21              ; ESAI

M_RHCKP   EQU     20               ;ESAI

M_RFSP   EQU     19              ; ESAI

M_RCKP   EQU     18               ;ESAI

M_RFP     EQU      $3C000        ;ESAI MASK

M_RFP3   EQU     17              ; ESAI

M_RFP2   EQU     16              ; ESAI

M_RFP1   EQU     15              ; ESAI

M_RFP0   EQU     14              ; ESAI

M_RDC     EQU     $3E00        ;ESAI MASK

M_RDC4   EQU     13              ; ESAI

M_RDC3   EQU     12              ; ESAI

M_RDC2   EQU     11              ; ESAI

M_RDC1   EQU     10              ; ESAI

M_RDC0   EQU     9              ; ESAI

M_RPSR   EQU     8               ; ESAI

M_RPM     EQU    $FF

M_RPM7   EQU     7               ; ESAI

M_RPM6   EQU     6               ; ESAI

M_RPM5   EQU     5               ; ESAI

M_RPM4   EQU     4               ; ESAI

M_RPM3   EQU     3               ; ESAI

M_RPM2   EQU     2               ; ESAI
```

```
M_RPM1   EQU     1               ; ESAI

M_RPM0   EQU     0               ; ESAI

;       RCR Register bits

M_RLIE   EQU     23              ; ESAI

M_RIE    EQU     22              ; ESAI

M_REDIE  EQU     21              ; ESAI

M_REIE   EQU     20              ; ESAI

M_RPR     EQU        19        ;          ESAI

M_RFSR   EQU     16              ; ESAI

M_RFSL   EQU     15              ; ESAI

M_RSWS    EQU       $7C00        ;ESAI MASK

M_RSWS4   EQU     14              ; ESAI

M_RSWS3   EQU     13              ; ESAI

M_RSWS2   EQU     12              ; ESAI

M_RSWS1   EQU     11              ; ESAI

M_RSWS0   EQU     10              ; ESAI

M_RMOD      EQU       $300

M_RMOD1  EQU     9               ; ESAI

M_RMOD0  EQU     8               ; ESAI

M_RWA   EQU     7               ; ESAI

M_RSHFD  EQU     6               ; ESAI

M_RE      EQU      $F

M_RE3    EQU     3               ; ESAI

M_RE2    EQU     2               ; ESAI

M_RE1    EQU     1               ; ESAI

M_RE0    EQU     0               ; ESAI

;       TCCR Register bits
```

```
M_THCKD  EQU     23              ; ESAI

M_TFSD   EQU     22              ; ESAI

M_TCKD   EQU     21              ; ESAI

M_THCKP   EQU     20               ;ESAI

M_TFSP   EQU     19              ; ESAI

M_TCKP   EQU     18              ; ESAI

M_TFP    EQU      $3C000

M_TFP3   EQU     17              ; ESAI

M_TFP2   EQU     16              ; ESAI

M_TFP1   EQU     15              ; ESAI

M_TFP0   EQU     14              ; ESAI

M_TDC     EQU     $3E00       ;

M_TDC4   EQU     13              ; ESAI

M_TDC3   EQU     12              ; ESAI

M_TDC2   EQU     11              ; ESAI

M_TDC1   EQU     10              ; ESAI

M_TDC0   EQU     9              ; ESAI

M_TPSR   EQU     8               ; ESAI

M_TPM     EQU    $FF            ;

M_TPM7   EQU     7               ; ESAI

M_TPM6   EQU     6               ; ESAI

M_TPM5   EQU     5               ; ESAI

M_TPM4   EQU     4               ; ESAI

M_TPM3   EQU     3               ; ESAI

M_TPM2   EQU     2               ; ESAI

M_TPM1   EQU     1               ; ESAI

M_TPM0   EQU     0               ; ESAI

;       TCR Register bits
```

```
M_TLIE   EQU     23              ; ESAI

M_TIE    EQU     22              ; ESAI

M_TEDIE  EQU     21              ; ESAI

M_TEIE   EQU     20              ; ESAI

M_TPR     EQU        19         ;          ESAI

M_PADC    EQU        17         ;          ESAI

M_TFSR   EQU     16              ; ESAI

M_TFSL   EQU     15              ; ESAI

M_TSWS   EQU    $7C00

M_TSWS4   EQU     14              ; ESAI

M_TSWS3   EQU     13              ; ESAI

M_TSWS2   EQU     12              ; ESAI

M_TSWS1   EQU     11              ; ESAI

M_TSWS0   EQU     10              ; ESAI

M_TMOD    EQU      $300

M_TMOD1  EQU     9               ; ESAI

M_TMOD0  EQU     8               ; ESAI

M_TWA   EQU     7               ; ESAI

M_TSHFD  EQU     6               ; ESAI

M_TEM       EQU     $3F

M_TE5    EQU     5               ; ESAI

M_TE4    EQU     4               ; ESAI

M_TE3    EQU     3               ; ESAI

M_TE2    EQU     2               ; ESAI

M_TE1    EQU     1               ; ESAI

M_TE0    EQU     0               ; ESAI

;       control bits of SAICR
```

```
M_ALC   EQU   8               ;ESAI

M_TEBE   EQU     7               ; ESAI

M_SYN    EQU     6               ; ESAI

M_OF2    EQU     2               ; ESAI

M_OF1    EQU     1               ; ESAI

M_OF0    EQU     0               ; ESAI

;       status bits of SAISR

M_TODE   EQU     17              ; ESAI

M_TEDE   EQU     16              ; ESAI

M_TDE    EQU     15              ; ESAI

M_TUE    EQU     14              ; ESAI

M_TFS    EQU     13              ; ESAI

M_RODF   EQU     10              ; ESAI

M_REDF   EQU     9               ; ESAI

M_RDF    EQU     8               ; ESAI

M_ROE    EQU     7               ; ESAI

M_RFS    EQU     6               ; ESAI

M_IF2    EQU     2               ; ESAI

M_IF1    EQU     1               ; ESAI

M_IF0    EQU     0               ; ESAI

;---------------------------------------------------------------------

;

;       EQUATES for HDI08

;

;---------------------------------------------------------------------

;       Register Addresses


M_HOTX    EQU     $FFFFC7         ; HOST Transmit Register (HOTX)
```

```
M_HORX    EQU     $FFFFC6          ; HOST Receive Register (HORX)
M_HBAR    EQU     $FFFFC5          ; HOST Base Address Register (HBAR)
M_HPCR    EQU     $FFFFC4          ; HOST Port Control Register (HPCR)
M_HSR     EQU     $FFFFC3          ; HOST Status Register (HSR)
M_HCR     EQU     $FFFFC2          ; HOST Control Register (HCR)
;       HCR bits
M_HRIE    EQU     $0               ; HOST Receive interrupts Enable
M_HOTIE   EQU     $1               ; HOST Transmit Interrupt Enable
M_HCIE    EQU     $2               ; HOST Command Interrupt Enable
M_HF2     EQU     $3               ; HOST Flag 2
M_HF3     EQU     $4               ; HOST Flag 3
M_HODM0   EQU     $5               ; HOST DMA Mode Control Bit 0
M_HODM1   EQU     $6               ; HOST DMA Mode Control Bit 1
M_HODM2   EQU     $7               ; HOST DMA Mode Control Bit 2

;       HSR bits
M_HRDF    EQU     $0               ; HOST Receive Data Full
M_HOTDE   EQU     $1               ; HOST Receive Data Emptiy
M_HCP     EQU     $2               ; HOST Command Pending
M_HF0     EQU     $3               ; HOST Flag 0
M_HF1     EQU     $4               ; HOST Flag 1
M_DMA     EQU     $7               ; HOST DMA Status

;       HPCR bits
M_HGEN    EQU     $0               ; HOST Port Enable
M_HA8EN   EQU     $1               ; HOST Address 8 Enable
M_HA9EN   EQU     $2               ; HOST Address 9 Enable
```

```
M_HCSEN  EQU     $3              ; HOST Chip Select Enable
M_HREN   EQU     $4              ; HOST Request Enable
M_HAEN   EQU     $5              ; HOST Acknowledge Enable
M_HOEN    EQU     $6              ; HOST Enable
M_HROD    EQU     $8              ; HOST Request Open Dranin mode
M_HDSP   EQU     $9              ; HOST Data Strobe Polarity
M_HASP   EQU     $a              ; HOST Address Strobe Polarity
M_HMUX   EQU     $b              ; HOST Multiplexed bus select
M_HDDS  EQU     $c              ; HOST Double/Single Strobe select
M_HCSP   EQU     $d              ; HOST Chip Select Polarity
M_HRP    EQU     $e              ; HOST Request Polarity
M_HAP    EQU     $f              ; HOST Acknowledge Polarity
;                 HBAR  BITS
M_BA   EQU   $FF
M_BA10   EQU   7
M_BA9   EQU   6
M_BA8   EQU   5
M_BA7   EQU   4
M_BA6   EQU   3
M_BA5   EQU   2
M_BA4   EQU   1
M_BA3   EQU   0
;---------------------------------------------------------------------
;
;       EQUATES for TIMER
;
;---------------------------------------------------------------------
;       Register Addresses Of TIMER0
```

```
M_TCSR0  EQU     $FFFF8F         ; TIMER0 Control/Status Register
M_TLR0   EQU     $FFFF8E     ; TIMER0 Load Reg
M_TCPR0  EQU     $FFFF8D         ; TIMER0 Compare Register
M_TCR0   EQU     $FFFF8C     ; TIMER0 Count Register
;       Register Addresses Of TIMER1
M_TCSR1  EQU     $FFFF8B         ; TIMER1 Control/Status Register
M_TLR1   EQU     $FFFF8A     ; TIMER1 Load Reg
M_TCPR1  EQU     $FFFF89         ; TIMER1 Compare Register
M_TCR1   EQU     $FFFF88     ; TIMER1 Count Register
;       Register Addresses Of TIMER2
M_TCSR2  EQU     $FFFF87         ; TIMER2 Control/Status Register
M_TLR2   EQU     $FFFF86     ; TIMER2 Load Reg
M_TCPR2  EQU     $FFFF85         ; TIMER2 Compare Register
M_TCR2   EQU     $FFFF84     ; TIMER2 Count Register
M_TPLR   EQU     $FFFF83     ; TIMER Prescaler Load Register
M_TPCR   EQU     $FFFF82     ; TIMER Prescalar Count Register
;       Timer Control/Status Register Bit Flags
M_TE     EQU     0               ; Timer Enable
M_TOIE   EQU     1               ; Timer Overflow Interrupt Enable
M_TCIE   EQU     2               ; Timer Compare Interrupt Enable
M_TC     EQU     $F0             ; Timer Control Mask (TC0-TC3)
M_INV    EQU     8               ; Inverter Bit
M_TRM    EQU     9               ; Timer Restart Mode
M_DIR    EQU     11              ; Direction Bit
M_DI     EQU     12              ; Data Input
M_DO     EQU     13              ; Data Output
M_PCE    EQU     15      ; Prescaled Clock Enable
```

```
M_TOF    EQU     20              ; Timer Overflow Flag

M_TCF    EQU     21              ; Timer Compare Flag

;       Timer Prescaler Register Bit Flags

M_PS     EQU  $600000     ; Prescaler Source Mask

M_PS0    EQU  21

M_PS1    EQU  22

;   Timer Control Bits

M_TC0    EQU     4               ; Timer Control 0

M_TC1    EQU     5               ; Timer Control 1

M_TC2    EQU     6               ; Timer Control 2

M_TC3    EQU     7               ; Timer Control 3

;------------------ end of ioequ.asm ------------------------
```

NOTES

Appendix C JTAG BSDL

```
-- FILENAME : 56366TQFP_revA.bsdl
--
-- M O T O R O L A   S S D T   J T A G   S O F T W A R E
-- BSDL File Generated: Mon Jan 18 10:13:53 1999
--
-- Revision History:
--

entity DSP56366 is
          generic (PHYSICAL_PIN_MAP : string := "TQFP144");

          port (  TDO:out      bit;
                  TDI:in       bit;
                  TMS:in       bit;
                  TCK:in       bit;
                  SCK:inout    bit;
                 SDO0:inout    bit;
                 SDO1:inout    bit;
               SDOI23:inout    bit;
                PINIT:in       bit;
               SDOI32:inout    bit;
                 SVCC:linkage  bit_vector(0 to 1);
                 SGND:linkage  bit_vector(0 to 1);
               SDOI41:inout    bit;
               SDOI50:inout    bit;
                  FST:inout    bit;
                  FSR:inout    bit;
                 SCKT:inout    bit;
                 SCKR:inout    bit;
                HSCKT:inout    bit;
                HSCKR:inout    bit;
                 QVCC:linkage  bit_vector(0 to 3);
                 QGND:linkage  bit_vector(0 to 3);
                QVCCH:linkage  bit_vector(0 to 2);
                   HP:inout    bit_vector(0 to 15);
                  ADO:inout    bit;
                  ACI:inout    bit;
                  TIO:inout    bit;
                 HVCC:linkage  bit;
                 HGND:linkage  bit;
                  SS_:in       bit;
                HREQ_:inout    bit;
               RESET_:in       bit;
                 PVCC:linkage  bit;
                 PCAP:linkage  bit;
                 PGND:linkage  bit;
                   AA:out      bit_vector(0 to 2);
```

```
        CAS_:out      bit;
       EXTAL:in       bit;
        CVCC:linkage  bit_vector(0 to 1);
        CGND:linkage  bit_vector(0 to 1);
         TA_:in       bit;
         BR_:buffer   bit;
         BB_:inout    bit;
         WR_:out      bit;
         RD_:out      bit;
         BG_:in       bit;
           A:out      bit_vector(0 to 17);
        AVCC:linkage  bit_vector(0 to 2);
        AGND:linkage  bit_vector(0 to 3);
           D:inout    bit_vector(0 to 23);
        DVCC:linkage  bit_vector(0 to 3);
        DGND:linkage  bit_vector(0 to 3);
        MODD:in       bit;
        MODC:in       bit;
        MODB:in       bit;
        MODA:in       bit;
        MOSI:inout    bit;
         SDA:inout    bit;
    SDO41_1:inout    bit;
    SDO50_1:inout    bit;
      FST_1:inout    bit;
      FSR_1:inout    bit;
     SCKR_1:inout    bit;
     SCKT_1:inout    bit);

use STD_1149_1_1994.all;

attribute COMPONENT_CONFORMANCE of DSP56366 : entity is "STD_1149_1_1993";

attribute PIN_MAP of DSP56366 : entity is PHYSICAL_PIN_MAP;

constant TQFP144 : PIN_MAP_STRING :=
"SCK:       1, " &
"SS_:       2, " &
"HREQ_:     3, " &
"SDO0:      4, " &
"SDO1:      5, " &
"SDOI23:    6, " &
"SDOI32:    7, " &
"SVCC:     (8, 25), " &
"SGND:     (9, 26), " &
"SDOI41:    10, " &
"SDOI50:    11, " &
"FST:       12, " &
"FSR:       13, " &
"SCKT:      14, " &
"SCKR:      15, " &
"HSCKT:     16, " &
"HSCKR:     17, " &
"QVCC:     (18, 56, 91, 126), " &
"QGND:     (19, 54, 90, 127), " &
```

```
          "QVCCH:    (20, 49, 95), " &
          "HP:       (43, 42, 41, 40, 37, 36, 35, 34, 33, 32, 31, 22, 21, 30, 24, 23), " &
          "ADO:       27, " &
          "ACI:       28, " &
          "TIO:       29, " &
          "HVCC:      38, " &
          "HGND:      39, " &
          "RESET_:    44, " &
          "PVCC:      45, " &
          "PCAP:      46, " &
          "PGND:      47, " &
          "SDO50_1:   48, " &
          "FST_1:     50, " &
          "AA:       (70, 69, 51), " &
          "CAS_:      52, " &
          "SCKT_1:    53, " &
          "EXTAL:     55, " &
          "CVCC:     (57, 65), " &
          "CGND:     (58, 66), " &
          "FSR_1:     59, " &
          "SCKR_1:    60, " &
          "PINIT:     61, " &
          "TA_:       62, " &
          "BR_:       63, " &
          "BB_:       64, " &
          "WR_:       67, " &
          "RD_:       68, " &
          "BG_:       71, " &
          "A:        (72, 73, 76, 77, 78, 79, 82, 83, 84, 85, 88, 89, 92, 93, 94, 97, 98, 99), " &
          "AVCC:     (74, 80, 86), " &
          "AGND:     (75, 81, 87, 96), " &
          "D:        (100, 101, 102, 105, 106, 107, 108, 109, 110, 113, 114, 115, 116, 117, 118,
121,  " &
                           "122, 123, 124, 125, 128, 131, 132, 133), " &
          "DVCC:     (103, 111, 119, 129), " &
          "DGND:     (104, 112, 120, 130), " &
          "MODD:      134, " &
          "MODC:      135, " &
          "MODB:      136, " &
          "MODA:      137, " &
          "SDO41_1:   138, " &
          "TDO:       139, " &
          "TDI:       140, " &
          "TCK:       141, " &
          "TMS:       142, " &
          "MOSI:      143, " &
          "SDA:       144 ";

          attribute TAP_SCAN_IN    of  TDI : signal is true;
          attribute TAP_SCAN_OUT   of  TDO : signal is true;
          attribute TAP_SCAN_MODE  of  TMS : signal is true;
          attribute TAP_SCAN_CLOCK of  TCK : signal is (20.0e6, BOTH);

          attribute INSTRUCTION_LENGTH of DSP56366 : entity is 4;
```

```
attribute INSTRUCTION_OPCODE of DSP56366 : entity is
   "EXTEST       (0000)," &
   "SAMPLE       (0001)," &
   "IDCODE       (0010)," &
   "CLAMP        (0101)," &
   "HIGHZ        (0100)," &
   "ENABLE_ONCE  (0110)," &
   "DEBUG_REQUEST(0111)," &
   "BYPASS       (1111)";

attribute INSTRUCTION_CAPTURE of DSP56366 : entity is "0001";
attribute IDCODE_REGISTER   of DSP56366 : entity is
   "0000"          & -- version
   "000111"        & -- manufacturer's use
   "00010011111"   & -- sequence number
   "00000001110"   & -- manufacturer identity
   "1";              -- 1149.1 requirement


attribute REGISTER_ACCESS of DSP56366 : entity is
   "ONCE[8]   (ENABLE_ONCE,DEBUG_REQUEST)" ;

attribute BOUNDARY_LENGTH of DSP56366 : entity is 152;

attribute BOUNDARY_REGISTER of DSP56366 : entity is
-- num    cell   port   func           safe [ccell  dis  rslt]
   "0     (BC_1, *,        control,       1)," &
   "1     (BC_6, SDO41_1,  bidir,         X,      0,   1,   Z)," &
   "2     (BC_1, MODA,     input,         X)," &
   "3     (BC_1, MODB,     input,         X)," &
   "4     (BC_1, MODC,     input,         X)," &
   "5     (BC_1, MODD,     input,         X)," &
   "6     (BC_6, D(23),    bidir,         1,     15,   1,   Z)," &
   "7     (BC_6, D(22),    bidir,         X,     15,   1,   Z)," &
   "8     (BC_6, D(21),    bidir,         X,     15,   1,   Z)," &
   "9     (BC_6, D(20),    bidir,         X,     15,   1,   Z)," &
   "10    (BC_6, D(19),    bidir,         X,     15,   1,   Z)," &
   "11    (BC_6, D(18),    bidir,         X,     15,   1,   Z)," &
   "12    (BC_6, D(17),    bidir,         X,     15,   1,   Z)," &
   "13    (BC_6, D(16),    bidir,         X,     15,   1,   Z)," &
   "14    (BC_6, D(15),    bidir,         X,     15,   1,   Z)," &
   "15    (BC_1, *,        control,       1)," &
   "16    (BC_6, D(14),    bidir,         X,     15,   1,   Z)," &
   "17    (BC_6, D(13),    bidir,         X,     15,   1,   Z)," &
   "18    (BC_6, D(12),    bidir,         X,     15,   1,   Z)," &
   "19    (BC_6, D(11),    bidir,         X,     28,   1,   Z)," &
-- num    cell   port   func           safe [ccell  dis  rslt]
   "20    (BC_6, D(10),    bidir,         X,     28,   1,   Z)," &
   "21    (BC_6, D(9),     bidir,         X,     28,   1,   Z)," &
   "22    (BC_6, D(8),     bidir,         X,     28,   1,   Z)," &
   "23    (BC_6, D(7),     bidir,         X,     28,   1,   Z)," &
   "24    (BC_6, D(6),     bidir,         X,     28,   1,   Z)," &
   "25    (BC_6, D(5),     bidir,         X,     28,   1,   Z)," &
   "26    (BC_6, D(4),     bidir,         X,     28,   1,   Z)," &
   "27    (BC_6, D(3),     bidir,         X,     28,   1,   Z)," &
```

```
   "28    (BC_1, *,        control,       1)," &
   "29    (BC_6, D(2),     bidir,         X,     28,   1,   Z)," &
   "30    (BC_6, D(1),     bidir,         X,     28,   1,   Z)," &
   "31    (BC_6, D(0),     bidir,         X,     28,   1,   Z)," &
   "32    (BC_1, A(17),    output3,       X,     35,   1,   Z)," &
   "33    (BC_1, A(16),    output3,       X,     35,   1,   Z)," &
   "34    (BC_1, A(15),    output3,       X,     35,   1,   Z)," &
   "35    (BC_1, *,        control,       1)," &
   "36    (BC_1, A(14),    output3,       X,     35,   1,   Z)," &
   "37    (BC_1, A(13),    output3,       X,     35,   1,   Z)," &
   "38    (BC_1, A(12),    output3,       X,     35,   1,   Z)," &
   "39    (BC_1, A(11),    output3,       X,     35,   1,   Z)," &
-- num    cell   port   func           safe [ccell  dis  rslt]
   "40    (BC_1, A(10),    output3,       X,     35,   1,   Z)," &
   "41    (BC_1, A(9),     output3,       X,     35,   1,   Z)," &
   "42    (BC_1, A(8),     output3,       X,     45,   1,   Z)," &
   "43    (BC_1, A(7),     output3,       X,     45,   1,   Z)," &
   "44    (BC_1, A(6),     output3,       X,     45,   1,   Z)," &
   "45    (BC_1, *,        control,       1)," &
   "46    (BC_1, A(5),     output3,       X,     45,   1,   Z)," &
   "47    (BC_1, A(4),     output3,       X,     45,   1,   Z)," &
   "48    (BC_1, A(3),     output3,       X,     45,   1,   Z)," &
   "49    (BC_1, A(2),     output3,       X,     45,   1,   Z)," &
   "50    (BC_1, A(1),     output3,       X,     45,   1,   Z)," &
   "51    (BC_1, A(0),     output3,       X,     45,   1,   Z)," &
   "52    (BC_1, BG_,      input,         X)," &
   "53    (BC_1, *,        control,       1)," &
   "54    (BC_1, AA(0),    output3,       X,     53,   1,   Z)," &
   "55    (BC_1, *,        control,       1)," &
   "56    (BC_1, AA(1),    output3,       X,     55,   1,   Z)," &
   "57    (BC_1, RD_,      output3,       X,     68,   1,   Z)," &
   "58    (BC_1, WR_,      output3,       X,     68,   1,   Z)," &
   "59    (BC_1, *,        control,       1)," &
-- num    cell   port   func           safe [ccell  dis  rslt]
   "60    (BC_6, BB_,      bidir,         X,     59,   1,   Z)," &
   "61    (BC_1, BR_,      output2,       X)," &
   "62    (BC_1, TA_,      input,         X)," &
   "63    (BC_1, PINIT,    input,         X)," &
   "64    (BC_1, *,        control,       1)," &
   "65    (BC_6, SCKR_1,   bidir,         X,     64,   1,   Z)," &
   "66    (BC_1, *,        control,       1)," &
   "67    (BC_6, FSR_1,    bidir,         X,     66,   1,   Z)," &
   "68    (BC_1, *,        control,       1)," &
   "69    (BC_1, EXTAL,    input,         X)," &
   "70    (BC_1, *,        control,       1)," &
   "71    (BC_6, SCKT_1,   bidir,         X,     70,   1,   Z)," &
   "72    (BC_1, *,        control,       1)," &
   "73    (BC_1, CAS_,     output3,       X,     72,   1,   Z)," &
   "74    (BC_1, *,        control,       1)," &
   "75    (BC_1, AA(2),    output3,       X,     74,   1,   Z)," &
   "76    (BC_1, *,        control,       1)," &
   "77    (BC_6, FST_1,    bidir,         X,     76,   1,   Z)," &
   "78    (BC_1, *,        control,       1)," &
   "79    (BC_6, SDO50_1,  bidir,         X,     78,   1,   Z)," &
-- num    cell   port   func           safe [ccell  dis  rslt]
```

```
    "80    (BC_1, RESET_,   input,         X)," &
    "81    (BC_1, *,        control,       1)," &
    "82    (BC_6, HP(0),    bidir,         X,     81,   1,   Z)," &
    "83    (BC_1, *,        control,       1)," &
    "84    (BC_6, HP(1),    bidir,         X,     83,   1,   Z)," &
    "85    (BC_1, *,        control,       1)," &
    "86    (BC_6, HP(2),    bidir,         X,     85,   1,   Z)," &
    "87    (BC_1, *,        control,       1)," &
    "88    (BC_6, HP(3),    bidir,         X,     87,   1,   Z)," &
    "89    (BC_1, *,        control,       1)," &
    "90    (BC_6, HP(4),    bidir,         X,     89,   1,   Z)," &
    "91    (BC_1, *,        control,       1)," &
    "92    (BC_6, HP(5),    bidir,         X,     91,   1,   Z)," &
    "93    (BC_1, *,        control,       1)," &
    "94    (BC_6, HP(6),    bidir,         X,     93,   1,   Z)," &
    "95    (BC_1, *,        control,       1)," &
    "96    (BC_6, HP(7),    bidir,         X,     95,   1,   Z)," &
    "97    (BC_1, *,        control,       1)," &
    "98    (BC_6, HP(8),    bidir,         X,     97,   1,   Z)," &
    "99    (BC_1, *,        control,       1)," &
-- num    cell   port   func           safe [ccell  dis  rslt]
    "100   (BC_6, HP(9),    bidir,         X,     99,   1,   Z)," &
    "101   (BC_1, *,        control,       1)," &
    "102   (BC_6, HP(10),   bidir,         X,    101,   1,   Z)," &
    "103   (BC_1, *,        control,       1)," &
    "104   (BC_6, HP(13),   bidir,         X,    103,   1,   Z)," &
    "105   (BC_1, *,        control,       1)," &
    "106   (BC_6, TIO,      bidir,         X,    105,   1,   Z)," &
    "107   (BC_1, *,        control,       1)," &
    "108   (BC_6, ACI,      bidir,         X,    107,   1,   Z)," &
    "109   (BC_1, *,        control,       1)," &
    "110   (BC_6, ADO,      bidir,         X,    109,   1,   Z)," &
    "111   (BC_1, *,        control,       1)," &
    "112   (BC_6, HP(14),   bidir,         X,    111,   1,   Z)," &
    "113   (BC_1, *,        control,       1)," &
    "114   (BC_6, HP(15),   bidir,         X,    113,   1,   Z)," &
    "115   (BC_1, *,        control,       1)," &
    "116   (BC_6, HP(11),   bidir,         X,    115,   1,   Z)," &
    "117   (BC_1, *,        control,       1)," &
    "118   (BC_6, HP(12),   bidir,         X,    117,   1,   Z)," &
    "119   (BC_1, *,        control,       1)," &
-- num    cell   port   func           safe [ccell  dis  rslt]
    "120   (BC_6, HSCKR,    bidir,         X,    119,   1,   Z)," &
    "121   (BC_1, *,        control,       1)," &
    "122   (BC_6, HSCKT,    bidir,         X,    121,   1,   Z)," &
    "123   (BC_1, *,        control,       1)," &
    "124   (BC_6, SCKR,     bidir,         X,    123,   1,   Z)," &
    "125   (BC_1, *,        control,       1)," &
    "126   (BC_6, SCKT,     bidir,         X,    125,   1,   Z)," &
    "127   (BC_1, *,        control,       1)," &
    "128   (BC_6, FSR,      bidir,         X,    127,   1,   Z)," &
    "129   (BC_1, *,        control,       1)," &
    "130   (BC_6, FST,      bidir,         X,    129,   1,   Z)," &
    "131   (BC_1, *,        control,       1)," &
    "132   (BC_6, SDOI50,   bidir,         X,    131,   1,   Z)," &
```

```
            "133   (BC_1, *,        control,       1)," &
            "134   (BC_6, SDOI41,   bidir,         X,    133,   1,   Z)," &
            "135   (BC_1, *,        control,       1)," &
            "136   (BC_6, SDOI32,   bidir,         X,    135,   1,   Z)," &
            "137   (BC_1, *,        control,       1)," &
            "138   (BC_6, SDOI23,   bidir,         X,    137,   1,   Z)," &
            "139   (BC_1, *,        control,       1)," &
         -- num    cell   port   func           safe [ccell  dis  rslt]
            "140   (BC_6, SDO1,     bidir,         X,    139,   1,   Z)," &
            "141   (BC_1, *,        control,       1)," &
            "142   (BC_6, SDO0,     bidir,         X,    141,   1,   Z)," &
            "143   (BC_1, *,        control,       1)," &
            "144   (BC_6, HREQ_,    bidir,         X,    143,   1,   Z)," &
            "145   (BC_1, SS_,      input,         X)," &
            "146   (BC_1, *,        control,       1)," &
            "147   (BC_6, SCK,      bidir,         X,    146,   1,   Z)," &
            "148   (BC_1, *,        control,       1)," &
            "149   (BC_6, SDA,      bidir,         X,    148,   1,   Z)," &
            "150   (BC_1, *,        control,       1)," &
            "151   (BC_6, MOSI,     bidir,         X,    150,   1,   Z)";

end DSP56366;
```

NOTES

Appendix D Programmer's Reference

D.1 Introduction

This section has been compiled as a reference for programmers. It contains a table showing the addresses of all the DSPs memory-mapped peripherals, an interrupt address table, an interrupt exception priority table, a quick reference to the host interface, and programming sheets for the major programmable registers on the DSP.

D.1.1 Peripheral Addresses

Table D-1 lists the memory addresses of all on-chip peripherals.

D.1.2 Interrupt Addresses

Table D-2 lists the interrupt starting addresses and sources.

D.1.3 Interrupt Priorities

Table D-3 lists the priorities of specific interrupts within interrupt priority levels.

D.1.4 Host Interface Quick Reference

Table D-4 is a quick reference guide to the host interface (HDI08).

D.1.5 Programming Sheets

The remaining figures describe major programmable registers on the DSP56366.

D.2 Internal I/O Memory Map

Table D-1. Internal I/O Memory Map

Peripheral	Address	Register Name
IPR	X:$FFFFFF	INTERRUPT PRIORITY REGISTER CORE (IPR-C)
	X:$FFFFFE	INTERRUPT PRIORITY REGISTER PERIPHERAL (IPR-P)
PLL	X:$FFFFFD	PLL CONTROL REGISTER (PCTL)
ONCE	X:$FFFFFC	ONCE GDB REGISTER (OGDB)
BIU	X:$FFFFFB	BUS CONTROL REGISTER (BCR)
	X:$FFFFFA	DRAM CONTROL REGISTER (DCR)
	X:$FFFFF9	ADDRESS ATTRIBUTE REGISTER 0 (AAR0)
	X:$FFFFF8	ADDRESS ATTRIBUTE REGISTER 1 (AAR1)
	X:$FFFFF7	ADDRESS ATTRIBUTE REGISTER 2 (AAR2)
	X:$FFFFF6	ADDRESS ATTRIBUTE REGISTER 3 (AAR3) [pin not available]
	X:$FFFFF5	ID REGISTER (IDR)
DMA	X:$FFFFF4	DMA STATUS REGISTER (DSTR)
	X:$FFFFF3	DMA OFFSET REGISTER 0 (DOR0)
	X:$FFFFF2	DMA OFFSET REGISTER 1 (DOR1)
	X:$FFFFF1	DMA OFFSET REGISTER 2 (DOR2)
	X:$FFFFF0	DMA OFFSET REGISTER 3 (DOR3)
DMA0	X:$FFFFEF	DMA SOURCE ADDRESS REGISTER (DSR0)
	X:$FFFFEE	DMA DESTINATION ADDRESS REGISTER (DDR0)
	X:$FFFFED	DMA COUNTER (DCO0)
	X:$FFFFEC	DMA CONTROL REGISTER (DCR0)
DMA1	X:$FFFFEB	DMA SOURCE ADDRESS REGISTER (DSR1)
	X:$FFFFEA	DMA DESTINATION ADDRESS REGISTER (DDR1)
	X:$FFFFE9	DMA COUNTER (DCO1)
	X:$FFFFE8	DMA CONTROL REGISTER (DCR1)
DMA2	X:$FFFFE7	DMA SOURCE ADDRESS REGISTER (DSR2)
	X:$FFFFE6	DMA DESTINATION ADDRESS REGISTER (DDR2)
	X:$FFFFE5	DMA COUNTER (DCO2)
	X:$FFFFE4	DMA CONTROL REGISTER (DCR2)
DMA3	X:$FFFFE3	DMA SOURCE ADDRESS REGISTER (DSR3)
	X:$FFFFE2	DMA DESTINATION ADDRESS REGISTER (DDR3)
	X:$FFFFE1	DMA COUNTER (DCO3)
	X:$FFFFE0	DMA CONTROL REGISTER (DCR3)

Table D-1. Internal I/O Memory Map (continued)

Peripheral	Address	Register Name
DMA4	X:$FFFFDF	DMA SOURCE ADDRESS REGISTER (DSR4)
	X:$FFFFDE	DMA DESTINATION ADDRESS REGISTER (DDR4)
	X:$FFFFDD	DMA COUNTER (DCO4)
	X:$FFFFDC	DMA CONTROL REGISTER (DCR4)
DMA5	X:$FFFFDB	DMA SOURCE ADDRESS REGISTER (DSR5)
	X:$FFFFDA	DMA DESTINATION ADDRESS REGISTER (DDR5)
	X:$FFFFD9	DMA COUNTER (DCO5)
	X:$FFFFD8	DMA CONTROL REGISTER (DCR5)
PORT D	X:$FFFFD7	PORT D CONTROL REGISTER (PCRD)
	X:$FFFFD6	PORT D DIRECTION REGISTER (PRRD)
	X:$FFFFD5	PORT D DATA REGISTER (PDRD)
DAX	X:$FFFFD4	DAX STATUS REGISTER (XSTR)
	X:$FFFFD3	DAX AUDIO DATA REGISTER B (XADRB)
	X:$FFFFD2	DAX AUDIO DATA REGISTER A (XADRA)
	X:$FFFFD1	DAX NON-AUDIO DATA REGISTER (XNADR)
	X:$FFFFD0	DAX CONTROL REGISTER (XCTR)
	X:$FFFFCF	*Reserved*
	X:$FFFFCE	*Reserved*
	X:$FFFFCD	*Reserved*
	X:$FFFFCC	*Reserved*
	X:$FFFFCB	*Reserved*
	X:$FFFFCA	*Reserved*
PORT B	X:$FFFFC9	HOST PORT GPIO DATA REGISTER (HDR)
	X:$FFFFC8	HOST PORT GPIO DIRECTION REGISTER (HDDR)
HDI08	X:$FFFFC7	HOST TRANSMIT REGISTER (HOTX)
	X:$FFFFC6	HOST RECEIVE REGISTER (HORX)
	X:$FFFFC5	HOST BASE ADDRESS REGISTER (HBAR)
	X:$FFFFC4	HOST PORT CONTROL REGISTER (HPCR)
	X:$FFFFC3	HOST STATUS REGISTER (HSR)
	X:$FFFFC2	HOST CONTROL REGISTER (HCR)
	X:$FFFFC1	*Reserved*
	X:$FFFFC0	*Reserved*
PORT C	X:$FFFFBF	PORT C CONTROL REGISTER (PCRC)
	X:$FFFFBE	PORT C DIRECTION REGISTER (PRRC)
	X:$FFFFBD	PORT C GPIO DATA REGISTER (PDRC)

Table D-1. Internal I/O Memory Map (continued)

Peripheral	Address	Register Name
ESAI	X:$FFFFBC	ESAI RECEIVE SLOT MASK REGISTER B (RSMB)
	X:$FFFFBB	ESAI RECEIVE SLOT MASK REGISTER A (RSMA)
	X:$FFFFBA	ESAI TRANSMIT SLOT MASK REGISTER B (TSMB)
	X:$FFFFB9	ESAI TRANSMIT SLOT MASK REGISTER A (TSMA)
	X:$FFFFB8	ESAI RECEIVE CLOCK CONTROL REGISTER (RCCR)
	X:$FFFFB7	ESAI RECEIVE CONTROL REGISTER (RCR)
	X:$FFFFB6	ESAI TRANSMIT CLOCK CONTROL REGISTER (TCCR)
	X:$FFFFB5	ESAI TRANSMIT CONTROL REGISTER (TCR)
	X:$FFFFB4	ESAI COMMON CONTROL REGISTER (SAICR)
	X:$FFFFB3	ESAI STATUS REGISTER (SAISR)
	X:$FFFFB2	*Reserved*
	X:$FFFFB1	*Reserved*
	X:$FFFFB0	*Reserved*
	X:$FFFFAF	*Reserved*
	X:$FFFFAE	*Reserved*
	X:$FFFFAD	*Reserved*
	X:$FFFFAC	*Reserved*
	X:$FFFFAB	ESAI RECEIVE DATA REGISTER 3 (RX3)
	X:$FFFFAA	ESAI RECEIVE DATA REGISTER 2 (RX2)
	X:$FFFFA9	ESAI RECEIVE DATA REGISTER 1 (RX1)
	X:$FFFFA8	ESAI RECEIVE DATA REGISTER 0 (RX0)
	X:$FFFFA7	*Reserved*
	X:$FFFFA6	ESAI TIME SLOT REGISTER (TSR)
	X:$FFFFA5	ESAI TRANSMIT DATA REGISTER 5 (TX5)
	X:$FFFFA4	ESAI TRANSMIT DATA REGISTER 4 (TX4)
	X:$FFFFA3	ESAI TRANSMIT DATA REGISTER 3 (TX3)
	X:$FFFFA2	ESAI TRANSMIT DATA REGISTER 2 (TX2)
	X:$FFFFA1	ESAI TRANSMIT DATA REGISTER 1 (TX1)
	X:$FFFFA0	ESAI TRANSMIT DATA REGISTER 0 (TX0)
	X:$FFFF9F	*Reserved*
	X:$FFFF9E	*Reserved*
	X:$FFFF9D	*Reserved*
	X:$FFFF9C	*Reserved*
	X:$FFFF9B	*Reserved*
	X:$FFFF9A	*Reserved*

Table D-1. Internal I/O Memory Map (continued)

Peripheral	Address	Register Name
	X:$FFFF99	*Reserved*
	X:$FFFF98	*Reserved*
	X:$FFFF97	*Reserved*
	X:$FFFF96	*Reserved*
	X:$FFFF95	*Reserved*
SHI	X:$FFFF94	SHI RECEIVE FIFO (HRX)
	X:$FFFF93	SHI TRANSMIT REGISTER (HTX)
	X:$FFFF92	SHI I^2C SLAVE ADDRESS REGISTER (HSAR)
	X:$FFFF91	SHI CONTROL/STATUS REGISTER (HCSR)
	X:$FFFF90	SHI CLOCK CONTROL REGISTER (HCKR)
TRIPLE TIMER	X:$FFFF8F	TIMER 0 CONTROL/STATUS REGISTER (TCSR0)
	X:$FFFF8E	TIMER 0 LOAD REGISTER (TLR0)
	X:$FFFF8D	TIMER 0 COMPARE REGISTER (TCPR0)
	X:$FFFF8C	TIMER 0 COUNT REGISTER (TCR0)
	X:$FFFF8B	TIMER 1 CONTROL/STATUS REGISTER (TCSR1)
	X:$FFFF8A	TIMER 1 LOAD REGISTER (TLR1)
	X:$FFFF89	TIMER 1 COMPARE REGISTER (TCPR1)
	X:$FFFF88	TIMER 1 COUNT REGISTER (TCR1)
	X:$FFFF87	TIMER 2 CONTROL/STATUS REGISTER (TCSR2)
	X:$FFFF86	TIMER 2 LOAD REGISTER (TLR2)
	X:$FFFF85	TIMER 2 COMPARE REGISTER (TCPR2)
	X:$FFFF84	TIMER 2 COUNT REGISTER (TCR2)
	X:$FFFF83	TIMER PRESCALER LOAD REGISTER (TPLR)
	X:$FFFF82	TIMER PRESCALER COUNT REGISTER (TPCR)
	X:$FFFF81	*Reserved*
	X:$FFFF80	*Reserved*
ESAI MUX PIN CONTROL	Y:$FFFFAF	ESAI MUX PIN CONTROL REGISTER (EMUXR)
	Y:$FFFFAE	*Reserved*
	Y:$FFFFAD	*Reserved*
	Y:$FFFFAC	*Reserved*
	Y:$FFFFAB	*Reserved*
	Y:$FFFFAA	*Reserved*
	Y:$FFFFA9	*Reserved*
	Y:$FFFFA8	*Reserved*

Table D-1. Internal I/O Memory Map (continued)

Peripheral	Address	Register Name
	Y:$FFFFA7	*Reserved*
	Y:$FFFFA6	*Reserved*
	Y:$FFFFA5	*Reserved*
	Y:$FFFFA4	*Reserved*
	Y:$FFFFA3	*Reserved*
	Y:$FFFFA2	*Reserved*
	Y:$FFFFA1	*Reserved*
	Y:$FFFFA0	*Reserved*
PORT E	Y:$FFFF9F	PORT E CONTROL REGISTER (PCRE)
	Y:$FFFF9E	PORT E DIRECTION REGISTER(PRRE)
	Y:$FFFF9D	PORT E GPIO DATA REGISTER(PDRE)

Table D-1. Internal I/O Memory Map (continued)

Peripheral	Address	Register Name
ESAI_1	Y:$FFFF9C	ESAI_1 RECEIVE SLOT MASK REGISTER B (RSMB_1)
	Y:$FFFF9B	ESAI_1 RECEIVE SLOT MASK REGISTER A (RSMA_1)
	Y:$FFFF9A	ESAI_1 TRANSMIT SLOT MASK REGISTER B (TSMB_1)
	Y:$FFFF99	ESAI_1 TRANSMIT SLOT MASK REGISTER A (TSMA_1)
	Y:$FFFF98	ESAI_1 RECEIVE CLOCK CONTROL REGISTER (RCCR_1)
	Y:$FFFF97	ESAI_1 RECEIVE CONTROL REGISTER (RCR_1)
	Y:$FFFF96	ESAI_1 TRANSMIT CLOCK CONTROL REGISTER (TCCR_1)
	Y:$FFFF95	ESAI_1 TRANSMIT CONTROL REGISTER (TCR_1)
	Y:$FFFF94	ESAI_1 COMMON CONTROL REGISTER (SAICR_1)
	Y:$FFFF93	ESAI_1 STATUS REGISTER (SAISR_1)
	Y:$FFFF92	*Reserved*
	Y:$FFFF91	*Reserved*
	Y:$FFFF90	*Reserved*
	Y:$FFFF8F	*Reserved*
	Y:$FFFF8E	*Reserved*
	Y:$FFFF8D	*Reserved*
	Y:$FFFF8C	*Reserved*
	Y:$FFFF8B	ESAI_1 RECEIVE DATA REGISTER 3 (RX3_1)
	Y:$FFFF8A	ESAI_1 RECEIVE DATA REGISTER 2 (RX2_1)
	Y:$FFFF89	ESAI_1 RECEIVE DATA REGISTER 1 (RX1_1)
	Y:$FFFF88	ESAI_1 RECEIVE DATA REGISTER 0 (RX0_1)
	Y:$FFFF87	*Reserved*
	Y:$FFFF86	ESAI_1 TIME SLOT REGISTER (TSR_1)
	Y:$FFFF85	ESAI_1 TRANSMIT DATA REGISTER 5 (TX5_1)
	Y:$FFFF84	ESAI_1 TRANSMIT DATA REGISTER 4 (TX4_1)
	Y:$FFFF83	ESAI_1 TRANSMIT DATA REGISTER 3 (TX3_1)
	Y:$FFFF82	ESAI_1 TRANSMIT DATA REGISTER 2 (TX2_1)
	Y:$FFFF81	ESAI_1 TRANSMIT DATA REGISTER 1 (TX1_1)
	Y:$FFFF80	ESAI_1 TRANSMIT DATA REGISTER 0 (TX0_1)

D.3 Interrupt Vector Addresses

Table D-2. DSP56366 Interrupt Vectors

Interrupt Starting Address	Interrupt Priority Level Range	Interrupt Source
VBA:$00	3	Hardware $\overline{\text{RESET}}$
VBA:$02	3	Stack Error
VBA:$04	3	Illegal Instruction
VBA:$06	3	Debug Request Interrupt
VBA:$08	3	Trap
VBA:$0A	3	Non-Maskable Interrupt ($\overline{\text{NMI}}$)
VBA:$0C	3	*Reserved* For Future Level-3 Interrupt Source
VBA:$0E	3	*Reserved* For Future Level-3 Interrupt Source
VBA:$10	0 - 2	IRQA
VBA:$12	0 - 2	IRQB
VBA:$14	0 - 2	IRQC
VBA:$16	0 - 2	IRQD
VBA:$18	0 - 2	DMA Channel 0
VBA:$1A	0 - 2	DMA Channel 1
VBA:$1C	0 - 2	DMA Channel 2
VBA:$1E	0 - 2	DMA Channel 3
VBA:$20	0 - 2	DMA Channel 4
VBA:$22	0 - 2	DMA Channel 5
VBA:$24	0 - 2	*Reserved*
VBA:$26	0 - 2	*Reserved*
VBA:$28	0 - 2	DAX Underrun Error
VBA:$2A	0 - 2	DAX Block Transferred
VBA:$2C	0 - 2	*Reserved*
VBA:$2E	0 - 2	DAX Audio Data Empty
VBA:$30	0 - 2	ESAI Receive Data
VBA:$32	0 - 2	ESAI Receive Even Data
VBA:$34	0 - 2	ESAI Receive Data With Exception Status
VBA:$36	0 - 2	ESAI Receive Last Slot
VBA:$38	0 - 2	ESAI Transmit Data
VBA:$3A	0 - 2	ESAI Transmit Even Data
VBA:$3C	0 - 2	ESAI Transmit Data with Exception Status
VBA:$3E	0 - 2	ESAI Transmit Last Slot
VBA:$40	0 - 2	SHI Transmit Data

Table D-2. DSP56366 Interrupt Vectors (continued)

Interrupt Starting Address	Interrupt Priority Level Range	Interrupt Source
VBA:$42	0 - 2	SHI Transmit Underrun Error
VBA:$44	0 - 2	SHI Receive FIFO Not Empty
VBA:$46	0 - 2	*Reserved*
VBA:$48	0 - 2	SHI Receive FIFO Full
VBA:$4A	0 - 2	SHI Receive Overrun Error
VBA:$4C	0 - 2	SHI Bus Error
VBA:$4E	0 - 2	*Reserved*
VBA:$50	0 - 2	*Reserved*
VBA:$52	0 - 2	*Reserved*
VBA:$54	0 - 2	TIMER0 Compare
VBA:$56	0 - 2	TIMER0 Overflow
VBA:$58	0 - 2	TIMER1 Compare
VBA:$5A	0 - 2	TIMER1 Overflow
VBA:$5C	0 - 2	TIMER2 Compare
VBA:$5E	0 - 2	TIMER2 Overflow
VBA:$60	0 - 2	Host Receive Data Full
VBA:$62	0 - 2	Host Transmit Data Empty
VBA:$64	0 - 2	Host Command (Default)
VBA:$66	0 - 2	*Reserved*
VBA:$68	0 - 2	*Reserved*
VBA:$6A	0 - 2	*Reserved*
VBA:$6C	0 - 2	*Reserved*
VBA:$6E	0 - 2	*Reserved*
VBA:$70	0 - 2	ESAI_1 Receive Data
VBA:$72	0 - 2	ESAI_1 Receive Even Data
VBA:$74	0 - 2	ESAI_1 Receive Data With Exception Status
VBA:$76	0 - 2	ESAI_1 Receive Last Slot
VBA:$78	0 - 2	ESAI_1 Transmit Data
VBA:$7A	0 - 2	ESAI_1 Transmit Even Data
VBA:$7C	0 - 2	ESAI_1 Transmit Data with Exception Status
VBA:$7E	0 - 2	ESAI_1 Transmit Last Slot
VBA:$80	0 - 2	*Reserved*
:	:	:
VBA:$FE	0 - 2	*Reserved*

D.4 Interrupt Source Priorities (within an IPL)

Table D-3. Interrupt Sources Priorities Within an IPL

Priority	Interrupt Source
Level 3 (Nonmaskable)	
Highest	Hardware $\overline{\text{RESET}}$
	Stack Error
	Illegal Instruction
	Debug Request Interrupt
	Trap
Lowest	Non-Maskable Interrupt
Levels 0, 1, 2 (Maskable)	
Highest	$\overline{\text{IRQA}}$ (External Interrupt)
	$\overline{\text{IRQB}}$ (External Interrupt)
	$\overline{\text{IRQC}}$ (External Interrupt)
	$\overline{\text{IRQD}}$ (External Interrupt)
	DMA Channel 0 Interrupt
	DMA Channel 1 Interrupt
	DMA Channel 2 Interrupt
	DMA Channel 3 Interrupt
	DMA Channel 4 Interrupt
	DMA Channel 5 Interrupt
	ESAI Receive Data with Exception Status
	ESAI Receive Even Data
	ESAI Receive Data
	ESAI Receive Last Slot
	ESAI Transmit Data with Exception Status
	ESAI Transmit Last Slot
	ESAI Transmit Even Data
	ESAI Transmit Data
	SHI Bus Error
	SHI Receive Overrun Error
	SHI Transmit Underrun Error
	SHI Receive FIFO Full
	SHI Transmit Data
	SHI Receive FIFO Not Empty
	HOST Command Interrupt
	HOST Receive Data Interrupt

Table D-3. Interrupt Sources Priorities Within an IPL (continued)

Priority	Interrupt Source
	HOST Transmit Data Interrupt
	DAX Transmit Underrun Error
	DAX Block Transferred
	DAX Transmit Register Empty
	TIMER0 Overflow Interrupt
	TIMER0 Compare Interrupt
	TIMER1 Overflow Interrupt
	TIMER1 Compare Interrupt
	TIMER2 Overflow Interrupt
	TIMER2 Compare Interrupt
	ESAI_1 Receive Data with Exception Status
	ESAI_1 Receive Even Data
	ESAI_1 Receive Data
	ESAI_1 Receive Last Slot
	ESAI_1 Transmit Data with Exception Status
	ESAI_1 Transmit Last Slot
	ESAI_1 Transmit Even Data
Lowest	ESAI_1 Transmit Data

D.5 Host Interface—Quick Reference

Table D-4. HDI08 Programming Model

Reg	Bit					Comments	Reset Type		
	Num	Mnemonic	Name	Val	Function		HW / SW	IR	ST
DSP SIDE									
HCR	0	HRIE	Receive Interrupt Enable	0 1	HRRQ interrupt disabled HRRQ interrupt enabled		0	-	-
	1	HTIE	Transmit Interrupt Enable	0 1	HTRQ interrupt disabled HTRQ interrupt enabled		0	-	-
	2	HCIE	Host Command Interrupt Enable	0 1	HCP interrupt disabled HCP interrupt enabled		0	-	-
	3	HF2	Host Flag 2				0		
	4	HF3	Host Flag 3				0	-	-
	7-5	HDM[2:0]	Host DMA Mode	000 100 001 010 011 101 110 111	DMA operation disabled DMA operation enabled 24-bit host-to-DSP DMA enabled 16-bit host-to-DSP DMA enabled 8-bit host-to-DSP DMA enabled 24-bit DSP-to-host DMA enabled 16-bit DSP-to-host DMA enabled 8-bit DSP-to-host DMA enabled		000		

Table D-4. HDI08 Programming Model (continued)

Reg	Bit					Comments	Reset Type		
	Num	Mnemonic	Name	Val	Function		HW / SW	IR	ST
HPCR	0	HGEN	Host GPIO Enable	0 1	GPIO pin disconnected GPIO pins active		0	-	-
	1	HA8EN	Host Address Line 8 Enable	0	HA8/HA1 = GPIO	this bit is treated as 1 if HMUX=0 this bit is treated as 0 if HEN=0	0	-	-
				1	HA8/HA1 = HA8/HA1				
	2	HA9EN	Host Address Line 9 Enable	0	HA9/HA2 = GPIO	this bit is treated as 1 if HMUX=0 this bit is treated as 0 if HEN=0	0	-	-
				1	HA9/HA2 = HA9/HA2				
	3	HCSEN	Host Chip Select Enable	0 1	HCS/HA10 = GPIO HCS/HA10 = HCS/HA10	this bit is treated as 0 if HEN=0	0	-	-
	4	HREN	Host Request Enable	0	HOREQ/HTRQ = GPIO HACK/HRRQ=GPIO	this bit is treated as 0 if HEN=0	0	-	-
				1	HOREQ/HTRQ=HOREQ/HTRQ HACK/HRRQ=HACK/HRRQ				
	5	HAEN	Host Acknowledge Enable	0	HACK/HRRQ = GPIO	this bit is ignored if HDRQ=1 this bit is treated as 0 if HREN=0 this bit is treated as 0 if HEN=0	0	-	-
				1	HACK/HRRQ= HACK				
	6	HEN	Host Enable	0 1	Host Port=GPIO Host Port Active		0	-	-
	8	HROD	Host Request Open Drain	0 1	HOREQ/HTRQ/HRRQ=driven HOREQ/HTRQ/HRRQ=open drain	this bit is ignored if HEN=0	0	-	-
	9	HDSP	Host Data Strobe Polarity	0 1	HDS/HRD/HWR active low HDS/HRD/HWR active high	this bit is ignored if HEN=0	0	-	-
	10	HASP	Host Address Strobe Polarity	0 1	HAS active low HAS active high	this bit is ignored if HEN=0	0	-	-
	11	HMUX	Host Multiplxed Bus	0 1	Separate address and data lines Multiplexed address/data	this bit is ignored if HEN=0	0	-	-
	12	HDDS	Host Dual Data Strobe	0 1	Single Data Strobe (HDS) Double Data Strobe (HWR, HRD)	this bit is ignored if HEN=0	0	-	-
	13	HCSP	Host Chip Select Polarity	0 1	HCS active low HCS active high	this bit is ignored if HEN=0	0	-	-
	14	HRP	Host Request polarity	0 1	HOREQ/HTRQ/HRRQ active low HOREQ/HTRQ/HRRQ active high	this bit is ignored if HEN=0	0	-	-
	15	HAP	Host Acknowledge Polarity	0 1	HACK active low HACK active high	this bit is ignored if HEN=0	0	-	-

Table D-4. HDI08 Programming Model (continued)

Reg	Bit					Comments	Reset Type		
	Num	Mnemonic	Name	Val	Function		HW / SW	IR	ST
HSR	0	HRDF	Host Receive Data Full	0 1	no receive data to be read receive data register is full		0	0	0
	1	HTDE	Host Transmit Data Empty	1 0	transmit data register empty transmit data reg. not empty		1	1	1
	2	HCP	Host Command Pending	0 1	no host command pending host command pending		0	0	0
	3	HF0	Host Flag0				0	-	-
	4	HF1	Host Flag1				0	-	-
	7	DMA	DMA Status	0 1	DMA mode disabled DMA mode enabled		0	-	-
HBAR	7-0	BA10-BA3	Host base Address Register				$80		
HORX	23-0		DSP Receive Data Register				empty		
HOTX	23-0		DSP Transmit Data Register				empty		
HDR	15-0	D15-D0	GPIO Pin Data				$0000	-	-
HDDR	15-0	DR15-DR0	GPIO Pin Direction	0 1	Input Output		$0000	-	-
Host Side									
ICR	0	RREQ	Receive Request Enable	0 1	HRRQ interrupt disabled HRRQ interrupt enabled		0	-	-
	1	TREQ	Transmit Request Enable	0 1	HTRQ interrupt disabled HTRQ interrupt enabled		0	-	-
	2	HDRQ	Double Host Request	0 1	HOREQ/HTRQ=HOREQ, HACK/HRRQ=HACK HOREQ/HTRQ=HTRQ, HACK/HRRQ=HRRQ	available if HDM2-HDM0=000	0	-	-
	3	HF0	Host Flag 0				0	-	-
	4	HF1	Host Flag 1				0	-	-
	5	HLEND	Host Little Endian	0 1	“Big Endian” order “Little Endian” order	available if HDM2-HDM0=000	0	-	-
	6-5	HM1-HM0	Host Mode Control	00 01 10 11	Interrupt Mode 24-bit DMA enabled 16-bit DMA enabled 8-bit DMA enabled	available if HDM2-HDM0=100	00	-	-
	7	INIT	Initialize	1	Reset data paths according to TREQ and RREQ	cleared by HDI08 hardware	0	-	-

Table D-4. HDI08 Programming Model (continued)

Reg	Bit					Comments	Reset Type		
	Num	Mnemonic	Name	Val	Function		HW / SW	IR	ST
ISR	0	RXDF	Receive Data Register Full	0 1	host receive register is empty host receive register is full		0	0	0
	1	TXDE	Transmit Data Register Empty	1 0	host transmit register empty host transmit register full		1	1	1
	2	TRDY	Transmitter Ready	1 0	transmit FIF O (6 deep) is empty transmit FIFO is not empty		1	1	1
	3	HF2	Host Flag2				0	-	-
	4	HF3	Host Flag3				0	-	-
	7	HREQ	Host Request	0 1	$\overline{\text{HOREQ}}$ pin is deasserted $\overline{\text{HOREQ}}$ pin is asserted (if enabled)		0	0	0
CVR	6-0	HV6-HV0	Host Command Vector			default vector	$2A	-	-
	7	HC	Host Command	0 1	no host command pending host command pending	cleared by HDI08 hardware when the HC int. req. is serviced	0	0	0
RXH/M/L	7-0		Host Receive Data Register				empty		
TXH/M/L	7-0		Host Transmit Data Register				empty		
IVR	7-0	IV7-IV0	Interrupt Register		68000 family vector register		$0F	-	-

D.6 Programming Sheets

The worksheets shown on the following pages contain listings of major programmable registers for the DSP56366. The programming sheets are grouped into the following order:

- Central Processor
- Host Interface (HDI08)
- Serial Host Interface (SHI)
- Two Enhanced Serial Audio Interfaces (ESAI and ESAI_1)
- Digital Audio Interface (DAX)
- Timer/Event Controller (TEC)
- GPIO (Ports B-E)

Each sheet provides room to write in the value of each bit and the hexadecimal value for each register. Programmers can photocopy these sheets and reuse them for each application development project.

For details on the instruction set of the DSP56300 family chips, see the *DSP56300 Family Manual*.

Application: _____ Date: _____

_____ Programmer: _____

Sheet 1 of 5

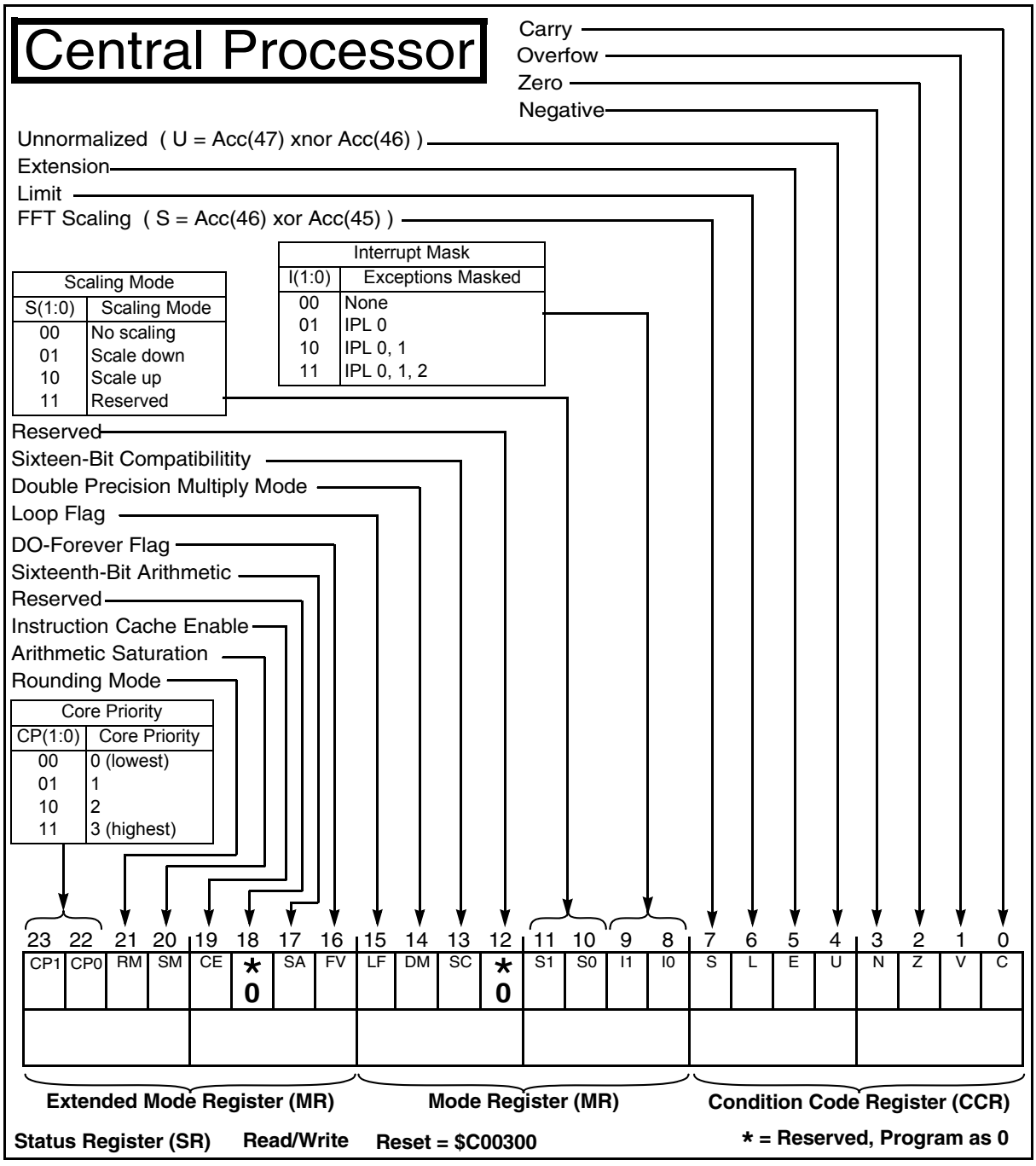


Figure D-1. Status Register (SR)

Application: _____ Date: _____

Programmer: _____

Sheet 2 of 5

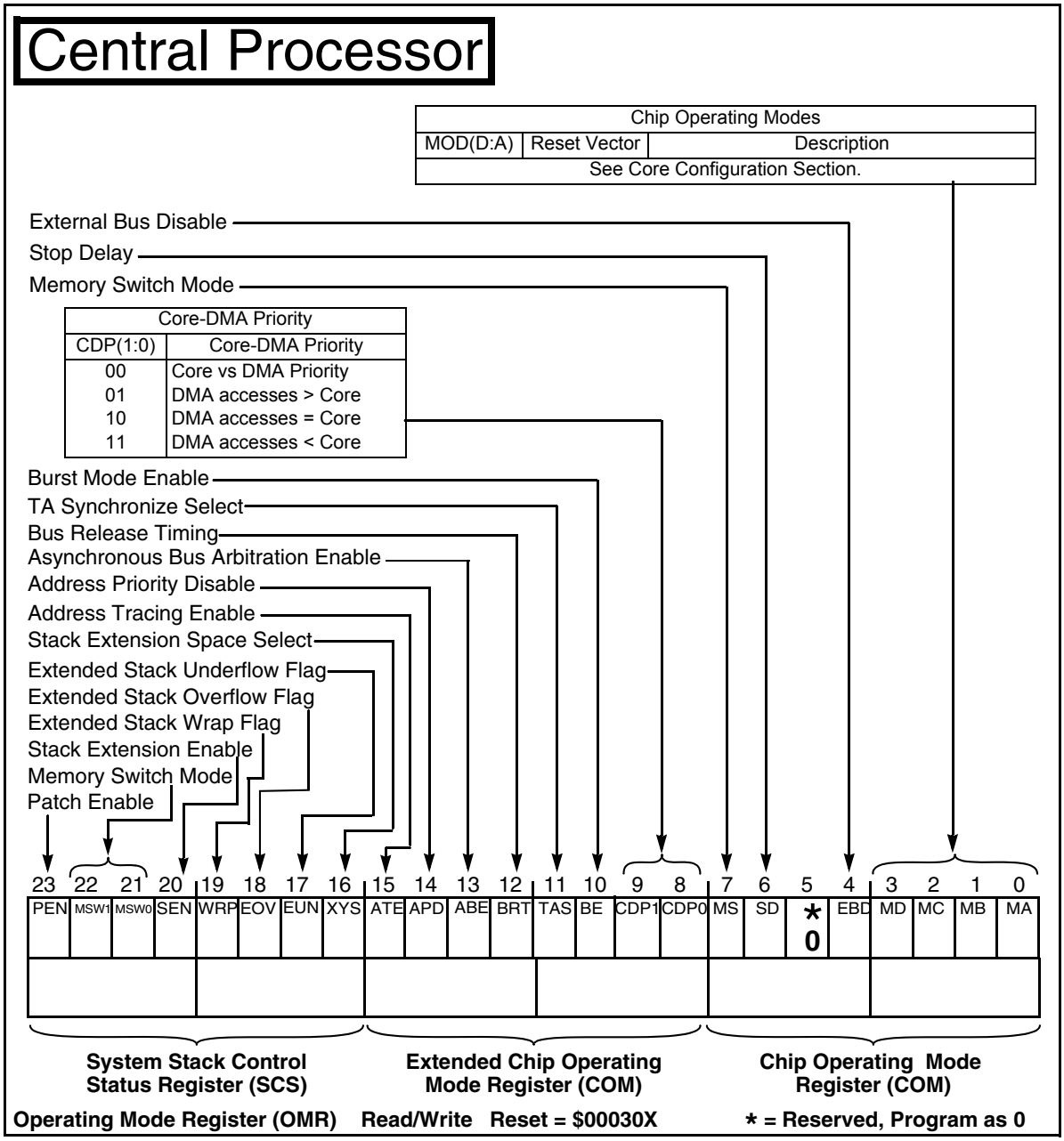

Central Processor
Chip Operating Modes
MOD(D:A) | Reset Vector | Description
See Core Configuration Section.
External Bus Disable
Stop Delay
Memory Switch Mode
Core-DMA Priority
CDP(1:0) | Core-DMA Priority
00 Core vs DMA Priority
01 DMA accesses > Core
10 DMA accesses = Core
11 DMA accesses < Core
Burst Mode Enable
TA Synchronize Select
Bus Release Timing
Asynchronous Bus Arbitration Enable
Address Priority Disable
Address Tracing Enable
Stack Extension Space Select
Extended Stack Underflow Flag
Extended Stack Overflow Flag
Extended Stack Wrap Flag
Stack Extension Enable
Memory Switch Mode
Patch Enable
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
PEN MSW1 MSW0 SEN WRP EOV EUN XYS ATE APD ABE BRT TAS BE CDP1 CDP0 MS SD * EBD MD MC MB MA
0
System Stack Control Status Register (SCS)
Extended Chip Operating Mode Register (COM)
Chip Operating Mode Register (COM)
Operating Mode Register (OMR) Read/Write Reset = $00030X
* = Reserved, Program as 0

Figure D-2. Operating Mode Register (OMR)

Application: _____ Date: _____

Programmer: _____

Sheet 3 of 5

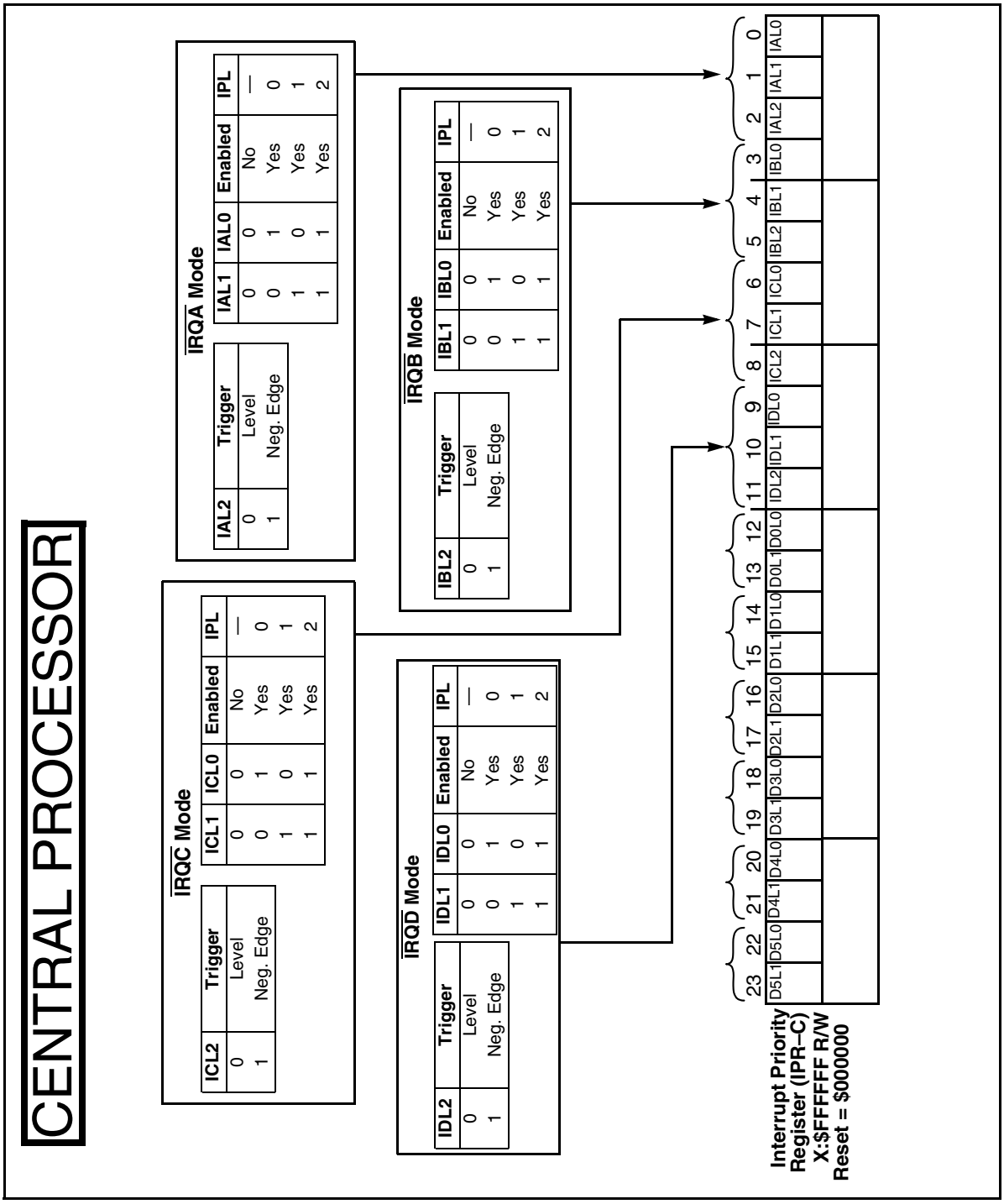


Figure D-3. Interrupt Priority Register–Core (IPR–C)

Application: _____ Date: _____

_____ Programmer: _____

Sheet 4 of 5

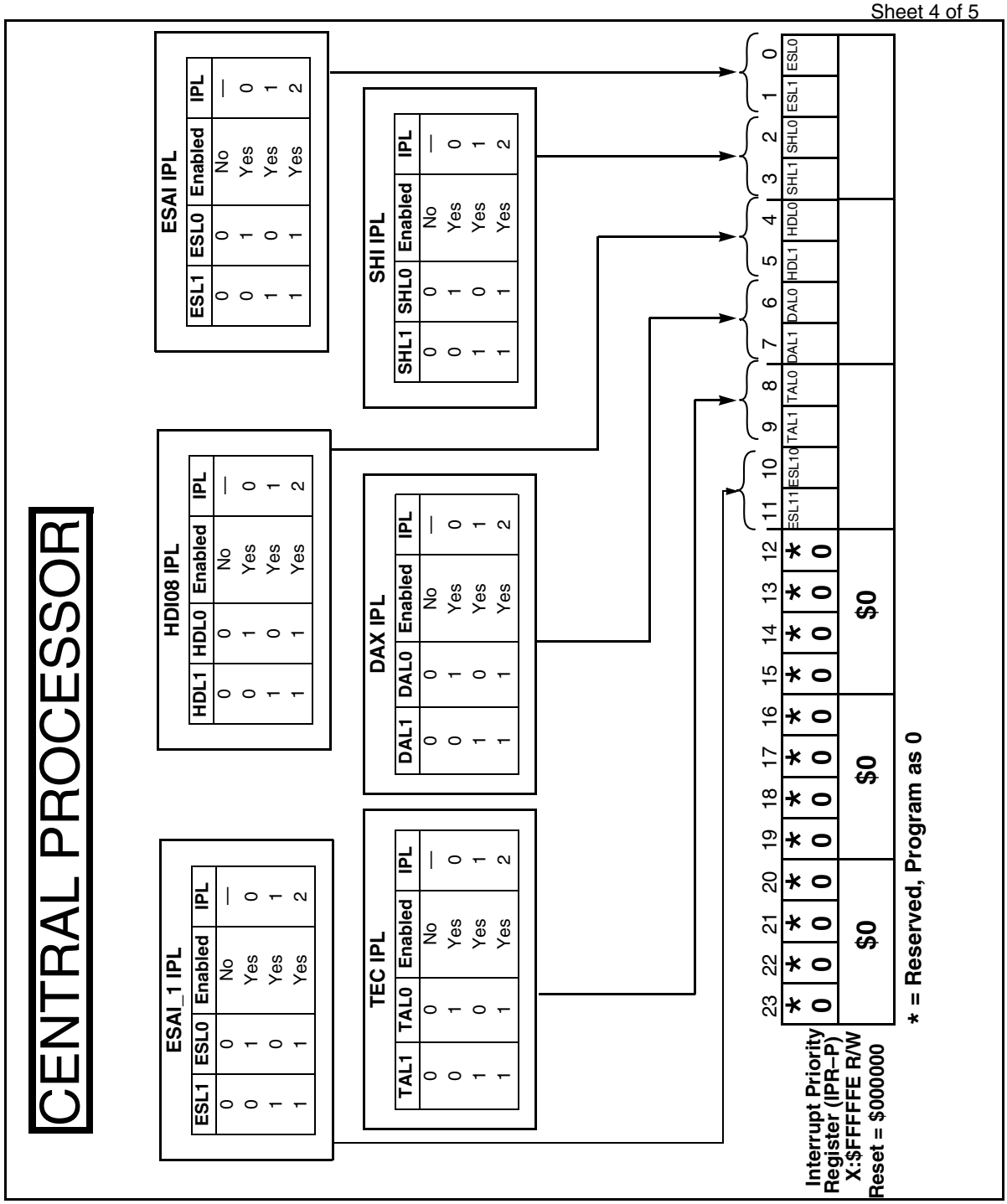

CENTRAL PROCESSOR
ESAI IPL
ESL1 ESL0 Enabled IPL
0 0 No —
0 1 Yes 0
1 0 Yes 1
1 1 Yes 2
SHI IPL
SHL1 SHL0 Enabled IPL
0 0 No —
0 1 Yes 0
1 0 Yes 1
1 1 Yes 2
HDI08 IPL
HDL1 HDL0 Enabled IPL
0 0 No —
0 1 Yes 0
1 0 Yes 1
1 1 Yes 2
DAX IPL
DAL1 DAL0 Enabled IPL
0 0 No —
0 1 Yes 0
1 0 Yes 1
1 1 Yes 2
TEC IPL
TAL1 TAL0 Enabled IPL
0 0 No —
0 1 Yes 0
1 0 Yes 1
1 1 Yes 2
ESAI_1 IPL
ESL1 ESL0 Enabled IPL
0 0 No —
0 1 Yes 0
1 0 Yes 1
1 1 Yes 2
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
* * * * * * * * * * * * ESL11 ESL10 TAL1 TAL0 DAL1 DAL0 HDL1 HDL0 SHL1 SHL0 ESL1 ESL0
0 0 0 0 0 0 0 0 0 0 0 0
$0 $0 $0
Interrupt Priority Register (IPR–P)
X:$FFFFFE R/W
Reset = $000000
* = Reserved, Program as 0

Figure D-4. Interrupt Priority Register – Peripherals (IPR–P)

Application: _____ Date: _____

_____ Programmer: _____

Sheet 5 of 5

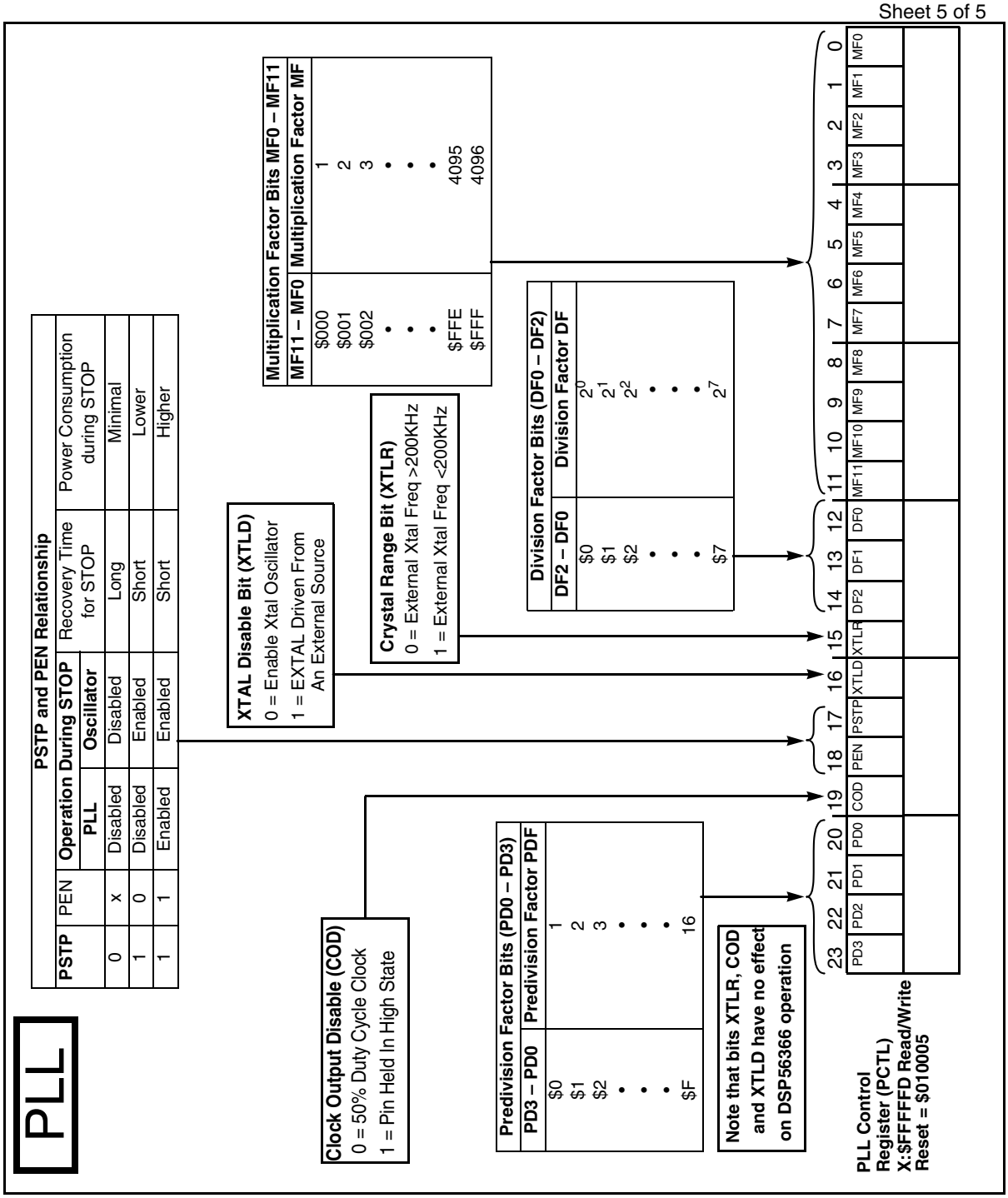

PLL
PSTP and PEN Relationship
PSTP | PEN | Operation During STOP: PLL | Operation During STOP: Oscillator | Recovery Time for STOP | Power Consumption during STOP
0 | x | Disabled | Disabled | Long | Minimal
1 | 0 | Disabled | Enabled | Short | Lower
1 | 1 | Enabled | Enabled | Short | Higher
XTAL Disable Bit (XTLD)
0 = Enable Xtal Oscillator
1 = EXTAL Driven From An External Source
Crystal Range Bit (XTLR)
0 = External Xtal Freq >200KHz
1 = External Xtal Freq <200KHz
Multiplication Factor Bits MF0 – MF11
MF11 – MF0 | Multiplication Factor MF
$000 | 1
$001 | 2
$002 | 3
$FFE | 4095
$FFF | 4096
Division Factor Bits (DF0 – DF2)
DF2 – DF0 | Division Factor DF
$0 | 2^0
$1 | 2^1
$2 | 2^2
$7 | 2^7
Clock Output Disable (COD)
0 = 50% Duty Cycle Clock
1 = Pin Held In High State
Predivision Factor Bits (PD0 – PD3)
PD3 – PD0 | Predivision Factor PDF
$0 | 1
$1 | 2
$2 | 3
$F | 16
Note that bits XTLR, COD and XTLD have no effect on DSP56366 operation
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
PD3 PD2 PD1 PD0 COD PEN PSTP XTLD XTLR DF2 DF1 DF0 MF11 MF10 MF9 MF8 MF7 MF6 MF5 MF4 MF3 MF2 MF1 MF0
PLL Control Register (PCTL)
X:$FFFFFD Read/Write
Reset = $010005

Figure D-5. Phase Lock Loop Control Register (PCTL)

Application: _____ Date: _____

_____ Programmer: _____

Sheet 1 of 6

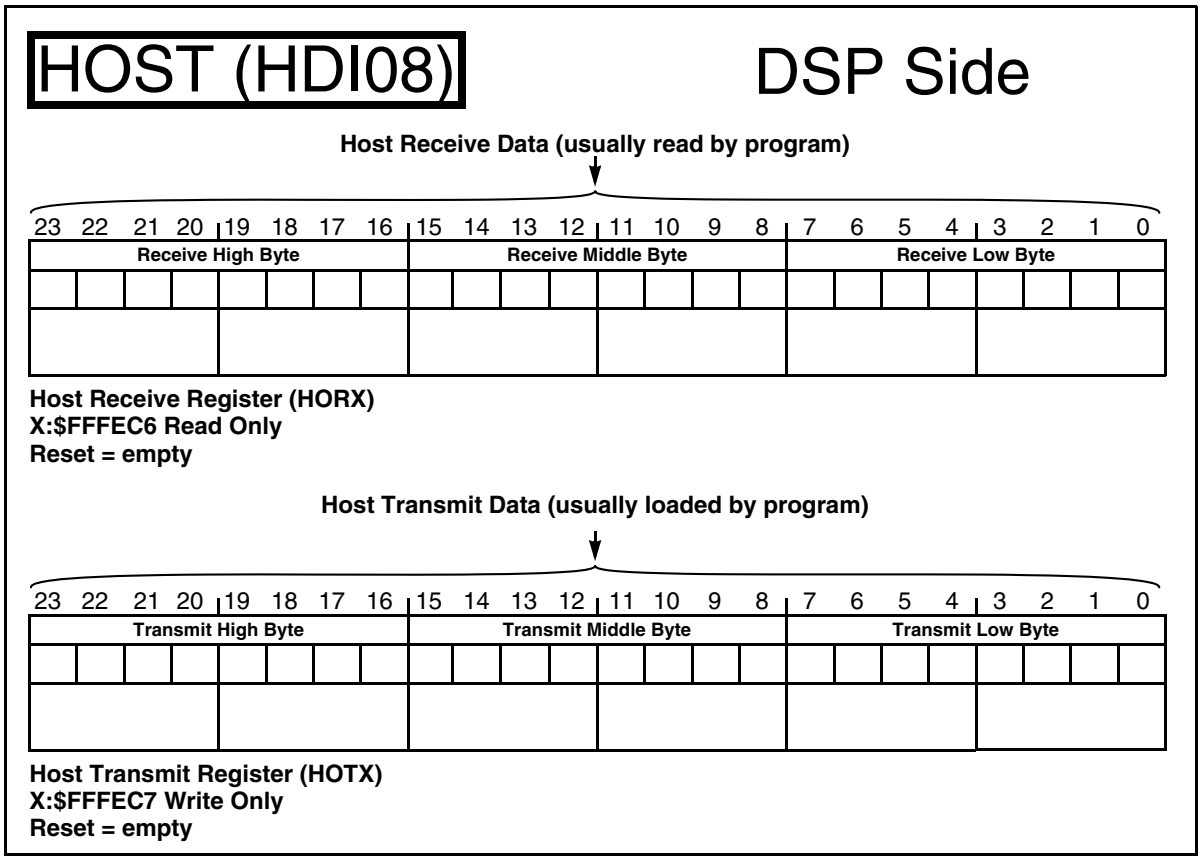

HOST (HDI08)
DSP Side
Host Receive Data (usually read by program)
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Receive High Byte
Receive Middle Byte
Receive Low Byte
Host Receive Register (HORX)
X:$FFFEC6 Read Only
Reset = empty
Host Transmit Data (usually loaded by program)
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Transmit High Byte
Transmit Middle Byte
Transmit Low Byte
Host Transmit Register (HOTX)
X:$FFFEC7 Write Only
Reset = empty

Figure D-6. Host Receive and Host Transmit Data Registers

Application: _____ Date: _____

_____ Programmer: _____

Sheet 2 of 6

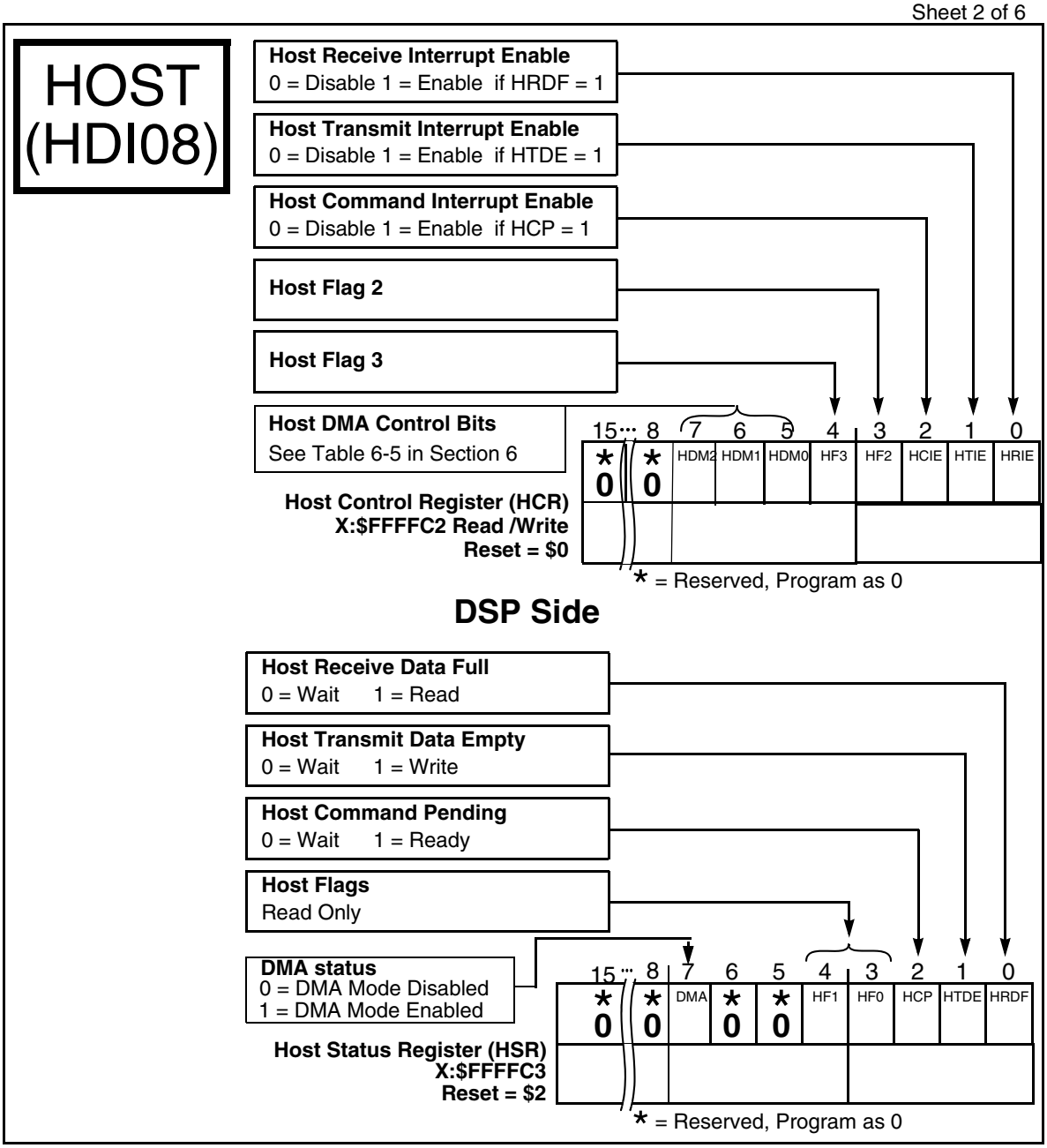

HOST
(HDI08)
Host Receive Interrupt Enable
0 = Disable 1 = Enable if HRDF = 1
Host Transmit Interrupt Enable
0 = Disable 1 = Enable if HTDE = 1
Host Command Interrupt Enable
0 = Disable 1 = Enable if HCP = 1
Host Flag 2
Host Flag 3
Host DMA Control Bits
See Table 6-5 in Section 6
15 ... 8 7 6 5 4 3 2 1 0
* * HDM2 HDM1 HDM0 HF3 HF2 HCIE HTIE HRIE
0 0
Host Control Register (HCR)
X:$FFFFC2 Read /Write
Reset = $0
* = Reserved, Program as 0
DSP Side
Host Receive Data Full
0 = Wait 1 = Read
Host Transmit Data Empty
0 = Wait 1 = Write
Host Command Pending
0 = Wait 1 = Ready
Host Flags
Read Only
DMA status
0 = DMA Mode Disabled
1 = DMA Mode Enabled
15 ... 8 7 6 5 4 3 2 1 0
* * DMA * * HF1 HF0 HCP HTDE HRDF
0 0 0 0
Host Status Register (HSR)
X:$FFFFC3
Reset = $2
* = Reserved, Program as 0

Figure D-7. Host Control and Status Registers

Application:_____ Date:_____

_____ Programmer:_____

Sheet 3 of 6

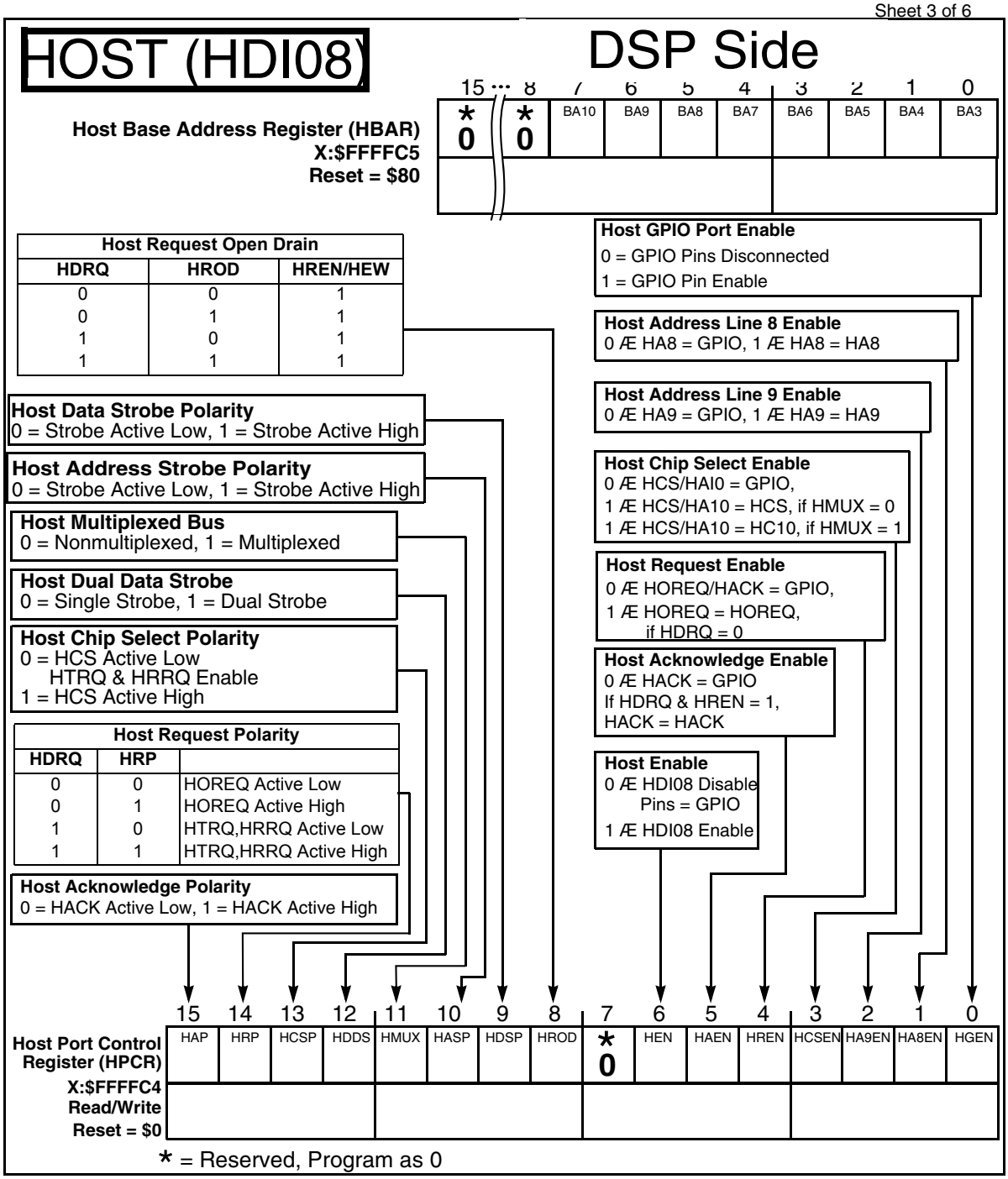

HOST (HDI08)
DSP Side
15 ... 8 7 6 5 4 3 2 1 0
* 0 * 0 BA10 BA9 BA8 BA7 BA6 BA5 BA4 BA3
Host Base Address Register (HBAR)
X:$FFFFC5
Reset = $80
Host Request Open Drain
HDRQ HROD HREN/HEW
0 0 1
0 1 1
1 0 1
1 1 1
Host GPIO Port Enable
0 = GPIO Pins Disconnected
1 = GPIO Pin Enable
Host Address Line 8 Enable
0 Æ HA8 = GPIO, 1 Æ HA8 = HA8
Host Address Line 9 Enable
0 Æ HA9 = GPIO, 1 Æ HA9 = HA9
Host Data Strobe Polarity
0 = Strobe Active Low, 1 = Strobe Active High
Host Address Strobe Polarity
0 = Strobe Active Low, 1 = Strobe Active High
Host Chip Select Enable
0 Æ HCS/HAI0 = GPIO,
1 Æ HCS/HA10 = HCS, if HMUX = 0
1 Æ HCS/HA10 = HC10, if HMUX = 1
Host Multiplexed Bus
0 = Nonmultiplexed, 1 = Multiplexed
Host Request Enable
0 Æ HOREQ/HACK = GPIO,
1 Æ HOREQ = HOREQ,
if HDRQ = 0
Host Dual Data Strobe
0 = Single Strobe, 1 = Dual Strobe
Host Acknowledge Enable
0 Æ HACK = GPIO
If HDRQ & HREN = 1,
HACK = HACK
Host Chip Select Polarity
0 = HCS Active Low
HTRQ & HRRQ Enable
1 = HCS Active High
Host Enable
0 Æ HDI08 Disable
Pins = GPIO
1 Æ HDI08 Enable
Host Request Polarity
HDRQ HRP
0 0 HOREQ Active Low
0 1 HOREQ Active High
1 0 HTRQ,HRRQ Active Low
1 1 HTRQ,HRRQ Active High
Host Acknowledge Polarity
0 = HACK Active Low, 1 = HACK Active High
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Host Port Control Register (HPCR)
HAP HRP HCSP HDDS HMUX HASP HDSP HROD * 0 HEN HAEN HREN HCSEN HA9EN HA8EN HGEN
X:$FFFFC4
Read/Write
Reset = $0
* = Reserved, Program as 0

Figure D-8. Host Base Address and Host Port Control

Application:_____ Date:_____

_____ Programmer:_____

Sheet 4 of 6

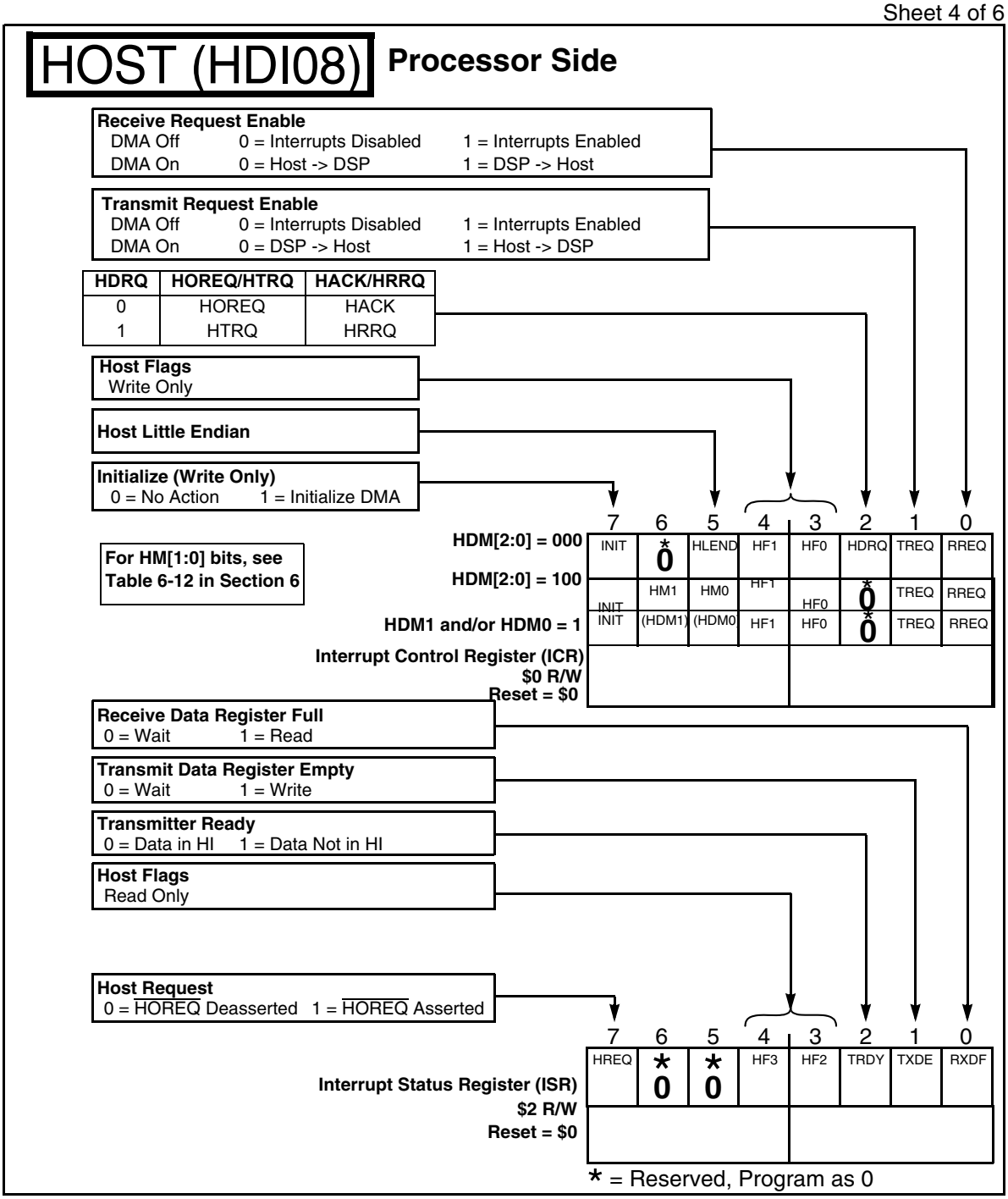

HOST (HDI08) Processor Side
Receive Request Enable
DMA Off 0 = Interrupts Disabled 1 = Interrupts Enabled
DMA On 0 = Host -> DSP 1 = DSP -> Host
Transmit Request Enable
DMA Off 0 = Interrupts Disabled 1 = Interrupts Enabled
DMA On 0 = DSP -> Host 1 = Host -> DSP
HDRQ HOREQ/HTRQ HACK/HRRQ
0 HOREQ HACK
1 HTRQ HRRQ
Host Flags
Write Only
Host Little Endian
Initialize (Write Only)
0 = No Action 1 = Initialize DMA
For HM[1:0] bits, see Table 6-12 in Section 6
HDM[2:0] = 000
HDM[2:0] = 100
HDM1 and/or HDM0 = 1
Interrupt Control Register (ICR)
$0 R/W
Reset = $0
7 6 5 4 3 2 1 0
INIT * 0 HLEND HF1 HF0 HDRQ TREQ RREQ
INIT HM1 HM0 HF1 HF0 * 0 TREQ RREQ
INIT (HDM1) (HDM0) HF1 HF0 * 0 TREQ RREQ
Receive Data Register Full
0 = Wait 1 = Read
Transmit Data Register Empty
0 = Wait 1 = Write
Transmitter Ready
0 = Data in HI 1 = Data Not in HI
Host Flags
Read Only
Host Request
0 = HOREQ Deasserted 1 = HOREQ Asserted
Interrupt Status Register (ISR)
$2 R/W
Reset = $0
7 6 5 4 3 2 1 0
HREQ * 0 * 0 HF3 HF2 TRDY TXDE RXDF
* = Reserved, Program as 0

Figure D-9. Host Interrupt Control and Interrupt Status

Application:_____ Date:_____

_____ Programmer:_____

Sheet 5 of 6

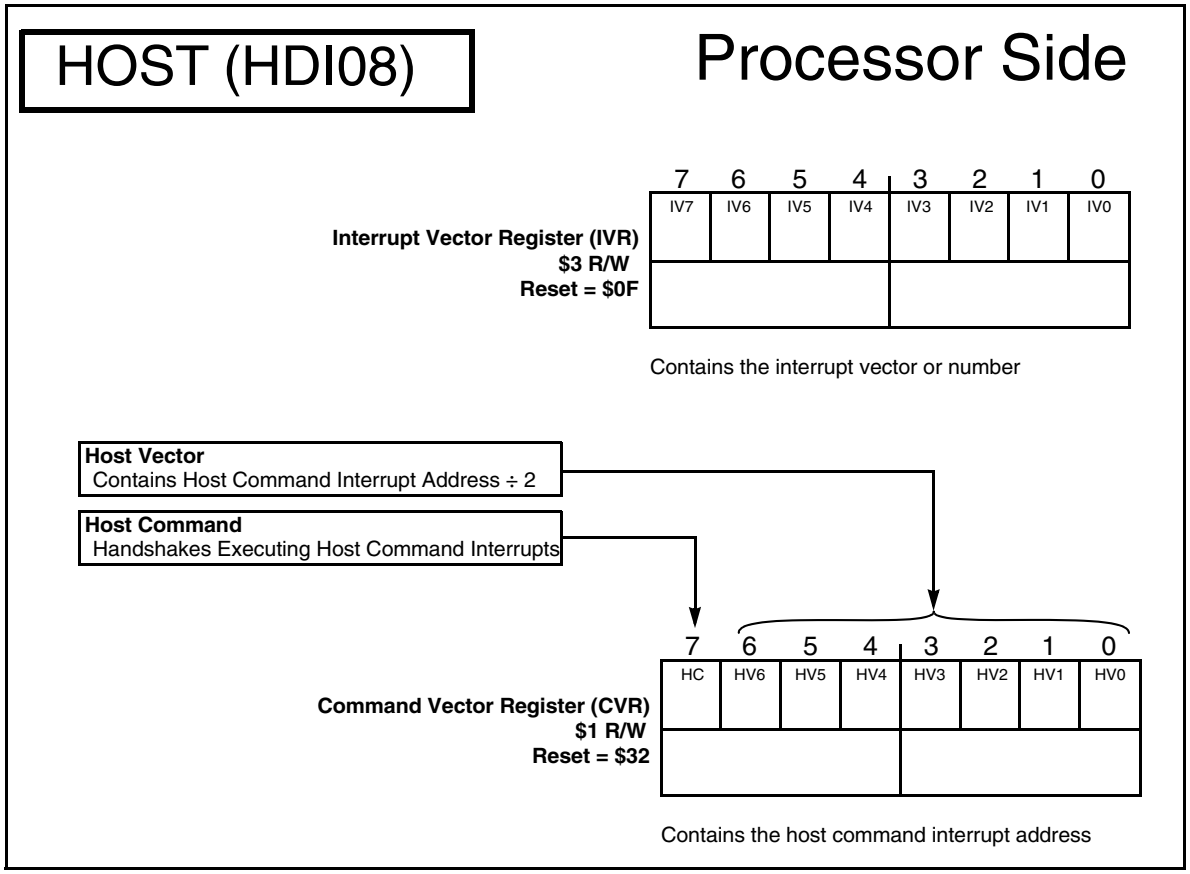

HOST (HDI08)
Processor Side
7 6 5 4 3 2 1 0
IV7 IV6 IV5 IV4 IV3 IV2 IV1 IV0
Interrupt Vector Register (IVR)
$3 R/W
Reset = $0F
Contains the interrupt vector or number
Host Vector
Contains Host Command Interrupt Address ÷ 2
Host Command
Handshakes Executing Host Command Interrupts
7 6 5 4 3 2 1 0
HC HV6 HV5 HV4 HV3 HV2 HV1 HV0
Command Vector Register (CVR)
$1 R/W
Reset = $32
Contains the host command interrupt address

Figure D-10. Host Interrupt Vector and Command Vector

Application:_____ Date:_____

_____ Programmer:_____

Sheet 6 of 6

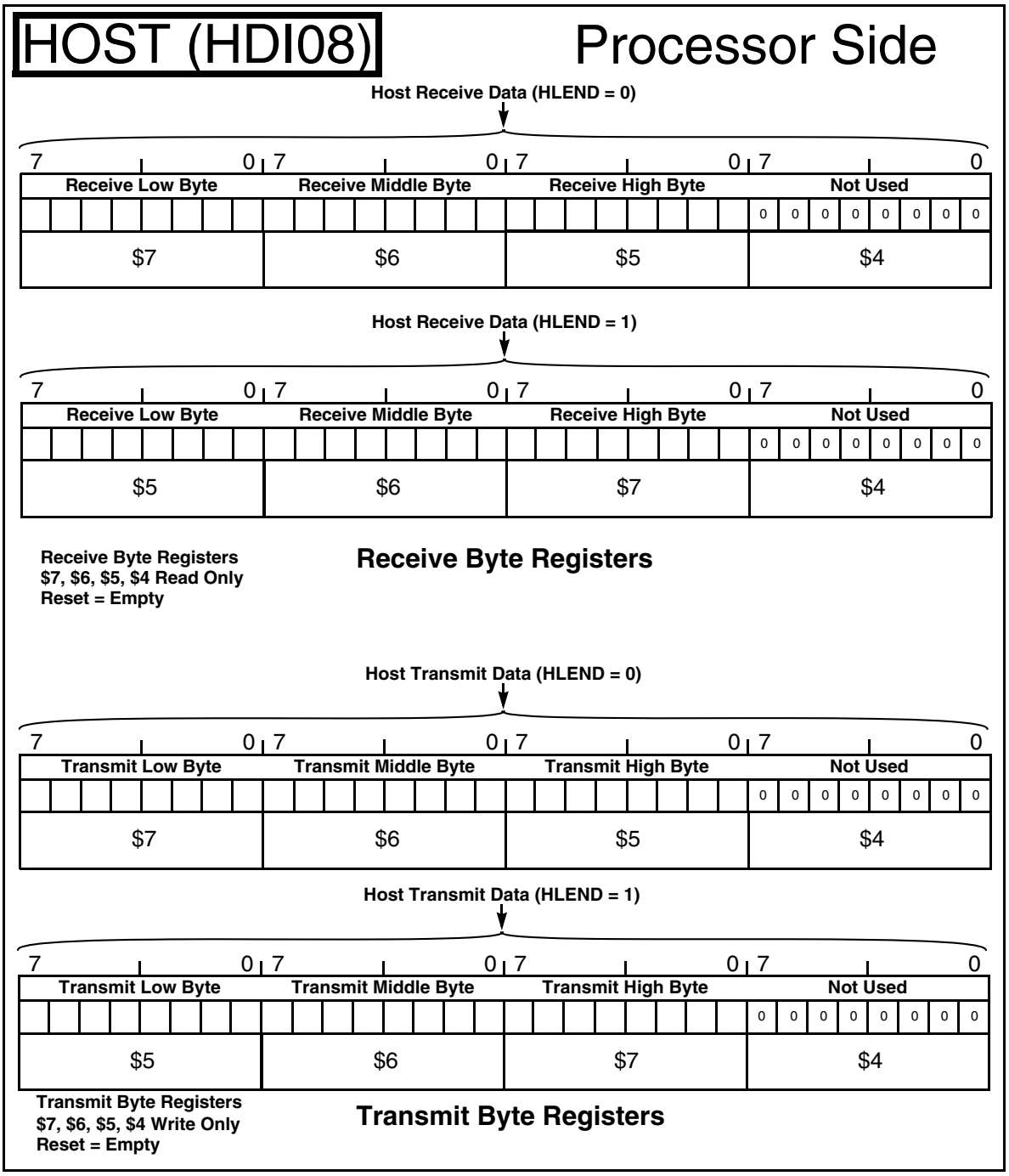


Figure D-11. Host Receive and Transmit Byte Registers

Application: _____ Date: _____

_____ Programmer: _____

Sheet 1 of 3

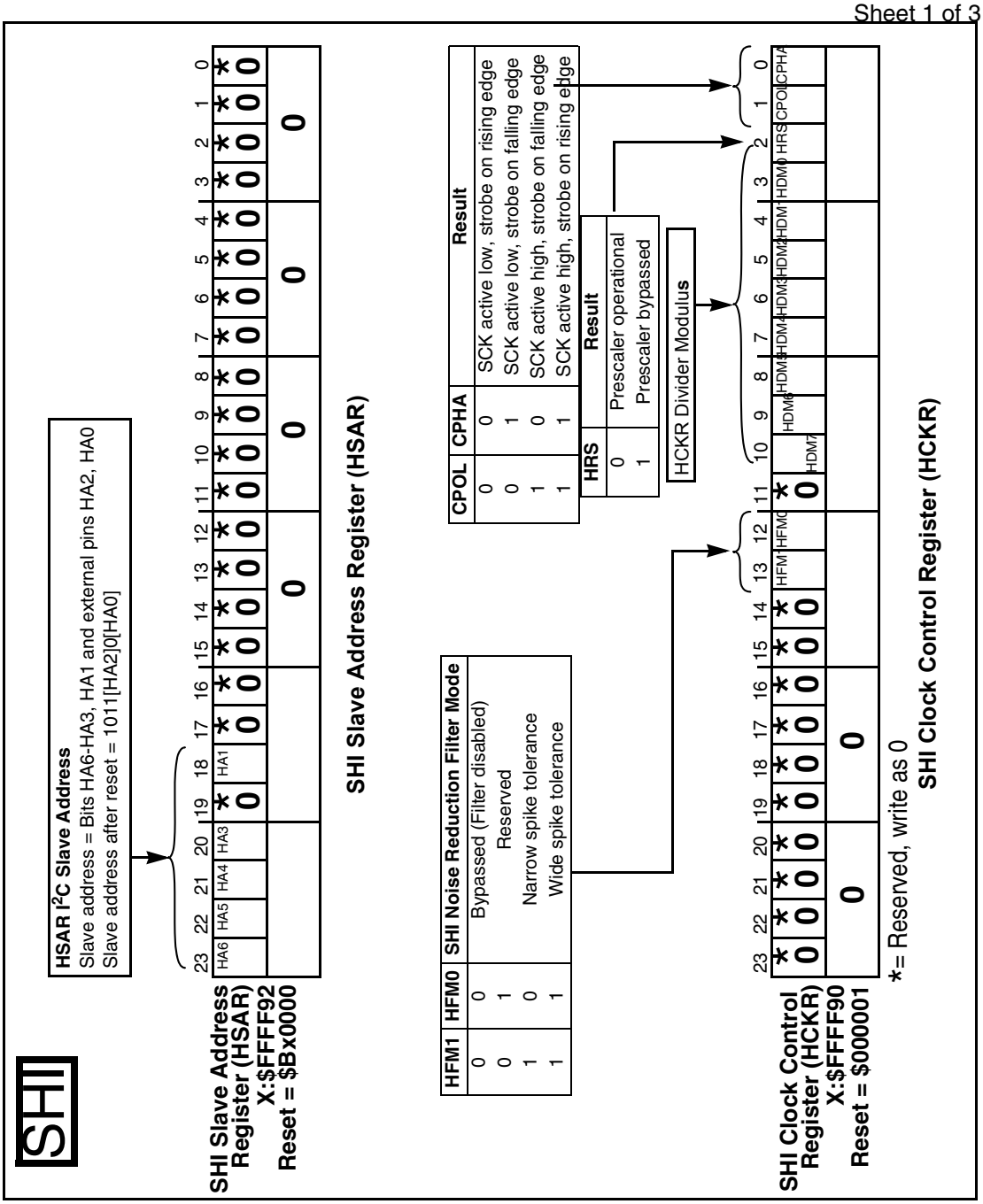

SHI
HSAR I2C Slave Address
Slave address = Bits HA6-HA3, HA1 and external pins HA2, HA0
Slave address after reset = 1011[HA2]0[HA0]
SHI Slave Address Register (HSAR)
X:$FFFF92
Reset = $Bx0000
SHI Slave Address Register (HSAR)
HFM1 HFM0 SHI Noise Reduction Filter Mode
0 0 Bypassed (Filter disabled)
0 1 Reserved
1 0 Narrow spike tolerance
1 1 Wide spike tolerance
CPOL CPHA Result
0 0 SCK active low, strobe on rising edge
0 1 SCK active low, strobe on falling edge
1 0 SCK active high, strobe on falling edge
1 1 SCK active high, strobe on rising edge
HRS Result
0 Prescaler operational
1 Prescaler bypassed
HCKR Divider Modulus
SHI Clock Control Register (HCKR)
X:$FFFF90
Reset = $000001
*= Reserved, write as 0
SHI Clock Control Register (HCKR)

Figure D-12. SHI Slave Address and Clock Control Registers

Application: _____ Date: _____

_____ Programmer: _____

Sheet 2 of 3

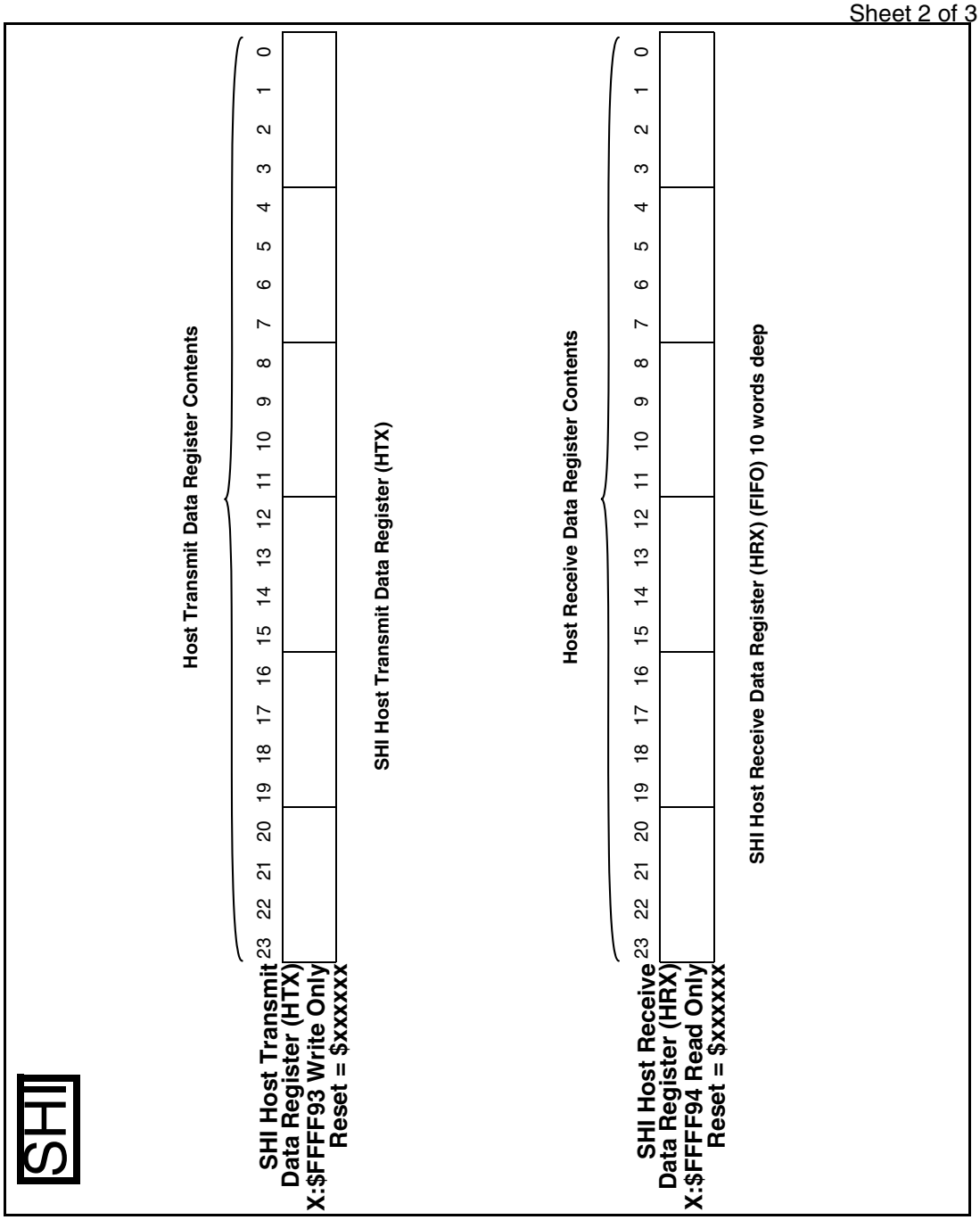

SHI
SHI Host Transmit Data Register (HTX)
X:$FFFF93 Write Only
Reset = $xxxxxx
Host Transmit Data Register Contents
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
SHI Host Transmit Data Register (HTX)
SHI Host Receive Data Register (HRX)
X:$FFFF94 Read Only
Reset = $xxxxxx
Host Receive Data Register Contents
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
SHI Host Receive Data Register (HRX) (FIFO) 10 words deep

Figure D-13. SHI Transmit and Receive Data Registers

Application:_____ Date:_____

Programmer:_____

Sheet 3 of 3

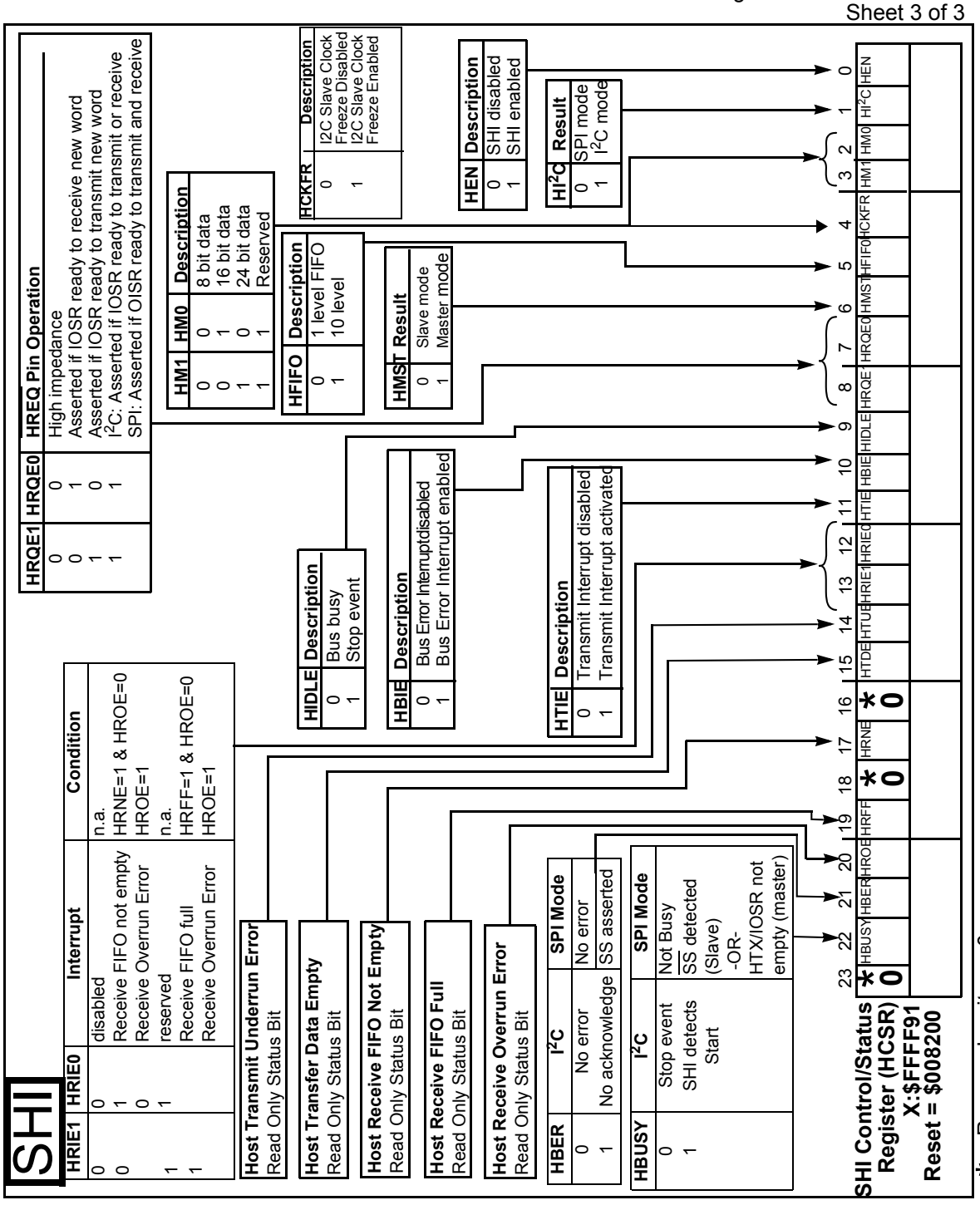

SHI
SHI Control/Status Register (HCSR)
X:$FFFF91
Reset = $008200
* = Reserved, write as 0
HRQE1 HRQE0 HREQ Pin Operation
0 0 High impedance
0 1 Asserted if IOSR ready to receive new word
1 0 Asserted if IOSR ready to transmit new word
1 1 I2C: Asserted if IOSR ready to transmit or receive
SPI: Asserted if OISR ready to transmit and receive
HRIE1 HRIE0 Interrupt Condition
0 0 disabled n.a.
0 1 Receive FIFO not empty HRNE=1 & HROE=0
1 0 Receive Overrun Error HROE=1
1 1 reserved n.a.
Receive FIFO full HRFF=1 & HROE=0
Receive Overrun Error HROE=1
HM1 HM0 Description
0 0 8 bit data
0 1 16 bit data
1 0 24 bit data
1 1 Reserved
HFIFO Description
0 1 level FIFO
1 10 level
HMST Result
0 Slave mode
1 Master mode
HCKFR Description
0 I2C Slave Clock Freeze Disabled
1 I2C Slave Clock Freeze Enabled
HEN Description
0 SHI disabled
1 SHI enabled
HI2C Result
0 SPI mode
1 I2C mode
HIDLE Description
0 Bus busy
1 Stop event
HBIE Description
0 Bus Error Interruptdisabled
1 Bus Error Interrupt enabled
HTIE Description
0 Transmit Interrupt disabled
1 Transmit Interrupt activated
Host Transmit Underrun Error
Read Only Status Bit
Host Transfer Data Empty
Read Only Status Bit
Host Receive FIFO Not Empty
Read Only Status Bit
Host Receive FIFO Full
Read Only Status Bit
Host Receive Overrun Error
Read Only Status Bit
HBER I2C SPI Mode
0 No error No error
1 No acknowledge SS asserted
HBUSY I2C SPI Mode
0 Stop event Not Busy
1 SHI detects Start SS detected (Slave) -OR- HTX/IOSR not empty (master)
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
* HBUSY HBER HROE HRFF * HRNE * HTDE HTUE HRIE1 HRIE0 HTIE HBIE HIDLE HRQE HRQE0 HMST HFIFO HCKFR HM1 HM0 HI2C HEN
0 0

Figure D-14. SHI Host Control/Status Register

Application: _____ Date: _____

Programmer: _____

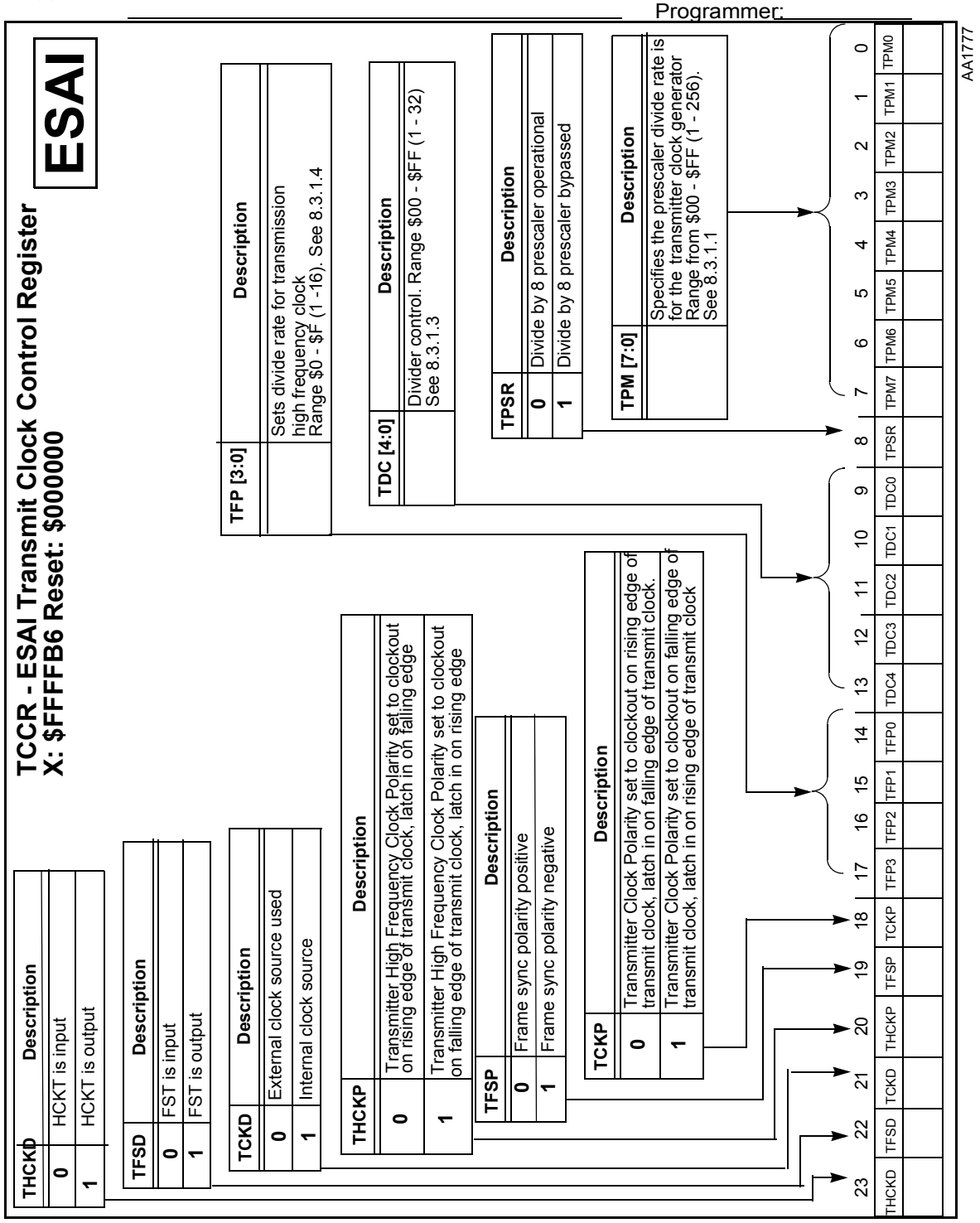

ESAI
TCCR - ESAI Transmit Clock Control Register
X: $FFFFB6 Reset: $000000
THCKD
0 HCKT is input
1 HCKT is output
TFSD
0 FST is input
1 FST is output
TCKD
0 External clock source used
1 Internal clock source
THCKP
0 Transmitter High Frequency Clock Polarity set to clockout on rising edge of transmit clock, latch in on falling edge
1 Transmitter High Frequency Clock Polarity set to clockout on falling edge of transmit clock, latch in on rising edge
TFSP
0 Frame sync polarity positive
1 Frame sync polarity negative
TCKP
0 Transmitter Clock Polarity set to clockout on rising edge of transmit clock, latch in on falling edge of transmit clock.
1 Transmitter Clock Polarity set to clockout on falling edge of transmit clock, latch in on rising edge of transmit clock
TFP [3:0]
Sets divide rate for transmission high frequency clock Range $0 - $F (1 -16). See 8.3.1.4
TDC [4:0]
Divider control. Range $00 - $FF (1 - 32) See 8.3.1.3
TPSR
0 Divide by 8 prescaler operational
1 Divide by 8 prescaler bypassed
TPM [7:0]
Specifies the prescaler divide rate is for the transmitter clock generator Range from $00 - $FF (1 - 256). See 8.3.1.1
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
THCKD TFSD TCKD THCKP TFSP TCKP TFP3 TFP2 TFP1 TFP0 TDC4 TDC3 TDC2 TDC1 TDC0 TPSR TPM7 TPM6 TPM5 TPM4 TPM3 TPM2 TPM1 TPM0
AA1777

Figure D-15. ESAI Transmit Clock Control Register

Application: _____ Date: _____

_____ Programmer: _____

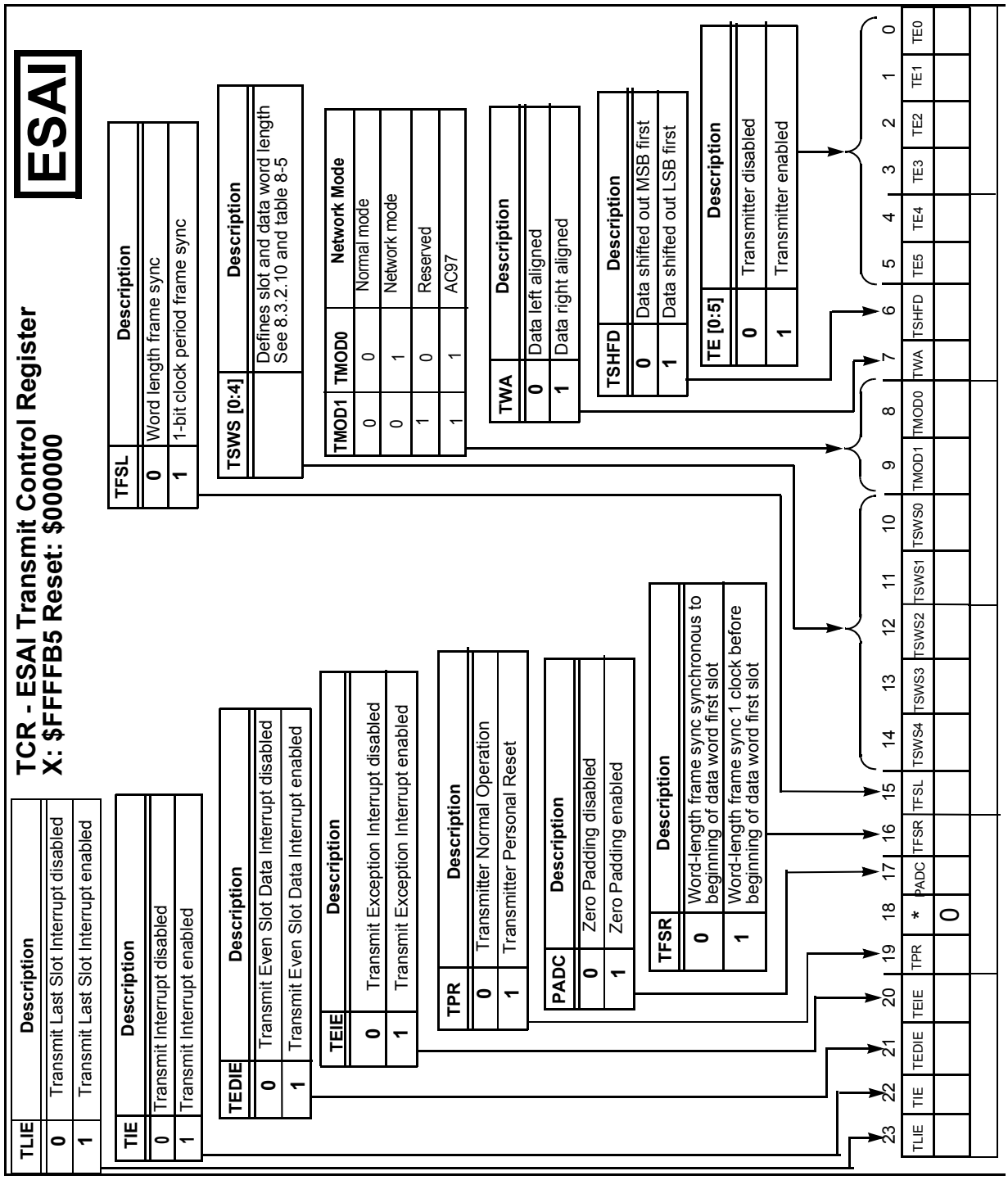


Figure D-16. ESAI Transmit Control Register

Application:_____ Date:_____

Programmer:_____

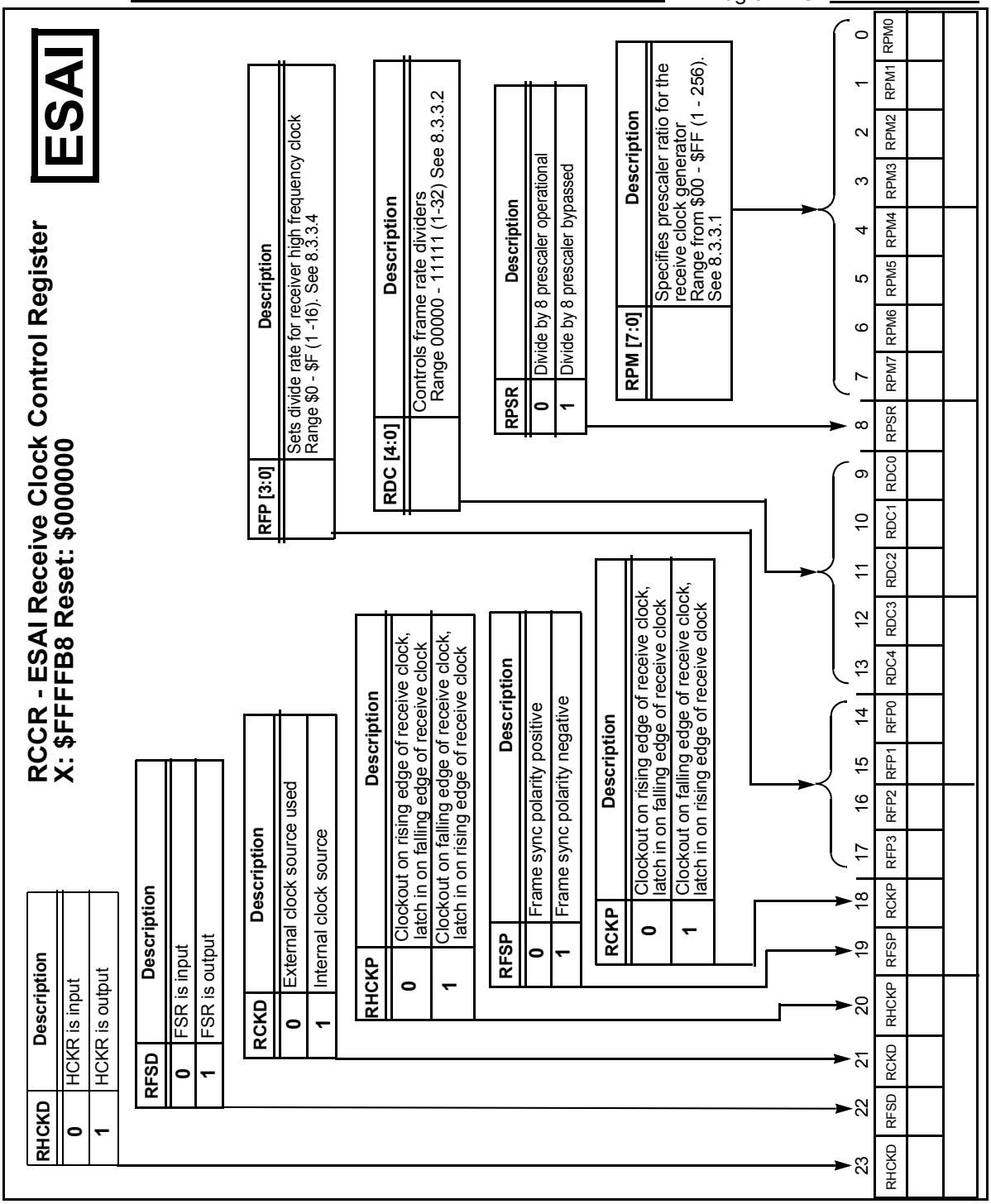

ESAI
RCCR - ESAI Receive Clock Control Register
X: $FFFFB8 Reset: $000000
RHCKD
Description
0
HCKR is input
1
HCKR is output
RFSD
Description
0
FSR is input
1
FSR is output
RCKD
Description
0
External clock source used
1
Internal clock source
RHCKP
Description
0
Clockout on rising edge of receive clock, latch in on falling edge of receive clock
1
Clockout on falling edge of receive clock, latch in on rising edge of receive clock
RFSP
Description
0
Frame sync polarity positive
1
Frame sync polarity negative
RCKP
Description
0
Clockout on rising edge of receive clock, latch in on falling edge of receive clock
1
Clockout on falling edge of receive clock, latch in on rising edge of receive clock
RFP [3:0]
Description
Sets divide rate for receiver high frequency clock Range $0 - $F (1 -16). See 8.3.3.4
RDC [4:0]
Description
Controls frame rate dividers Range 00000 - 11111 (1-32) See 8.3.3.2
RPSR
Description
0
Divide by 8 prescaler operational
1
Divide by 8 prescaler bypassed
RPM [7:0]
Description
Specifies prescaler ratio for the receive clock generator Range from $00 - $FF (1 - 256). See 8.3.3.1
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
RHCKD RFSD RCKD RHCKP RFSP RCKP RFP3 RFP2 RFP1 RFP0 RDC4 RDC3 RDC2 RDC1 RDC0 RPSR RPM7 RPM6 RPM5 RPM4 RPM3 RPM2 RPM1 RPM0

Figure D-17. ESAI Receive Clock Control Register

Application: _____ Date: _____

Programmer: _____

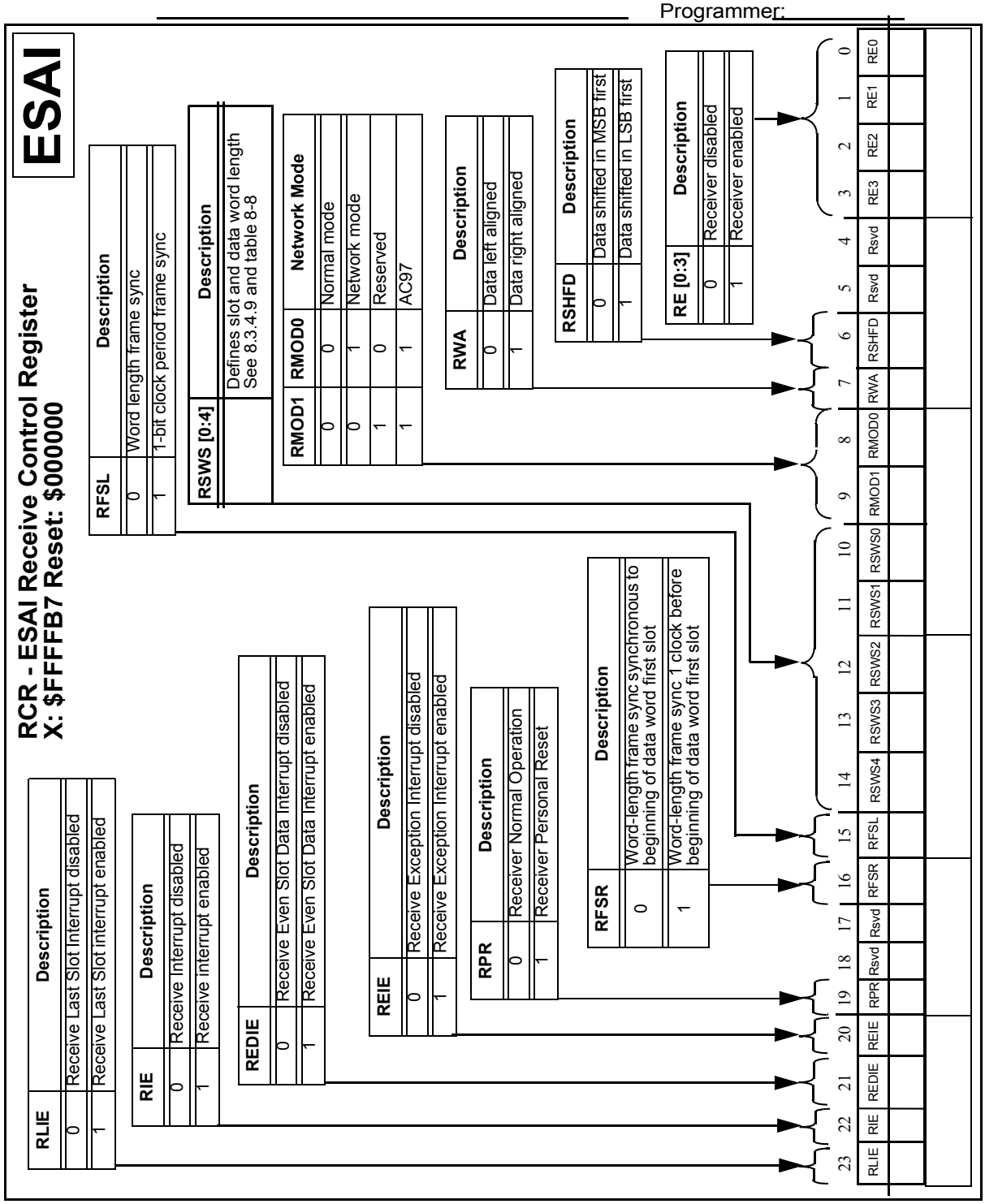

ESAI
RCR - ESAI Receive Control Register
X: $FFFFB7 Reset: $000000
RLIE
0 Receive Last Slot Interrupt disabled
1 Receive Last Slot interrupt enabled
RIE
0 Receive Interrupt disabled
1 Receive interrupt enabled
REDIE
0 Receive Even Slot Data Interrupt disabled
1 Receive Even Slot Data Interrupt enabled
REIE
0 Receive Exception Interrupt disabled
1 Receive Exception Interrupt enabled
RPR
0 Receiver Normal Operation
1 Receiver Personal Reset
RFSR
0 Word-length frame sync synchronous to beginning of data word first slot
1 Word-length frame sync 1 clock before beginning of data word first slot
RFSL
0 Word length frame sync
1 1-bit clock period frame sync
RSWS [0:4]
Defines slot and data word length See 8.3.4.9 and table 8-8
RMOD1 RMOD0 Network Mode
0 0 Normal mode
0 1 Network mode
1 0 Reserved
1 1 AC97
RWA
0 Data left aligned
1 Data right aligned
RSHFD
0 Data shifted in MSB first
1 Data shifted in LSB first
RE [0:3]
0 Receiver disabled
1 Receiver enabled
23 RLIE 22 RIE 21 REDIE 20 REIE 19 RPR 18 Rsvd 17 Rsvd 16 RFSR 15 RFSL 14 RSWS4 13 RSWS3 12 RSWS2 11 RSWS1 10 RSWS0 9 RMOD1 8 RMOD0 7 RWA 6 RSHFD 5 Rsvd 4 Rsvd 3 RE3 2 RE2 1 RE1 0 RE0

Figure D-18. ESAI Receive Control Register

Application: _____ Date: _____

Programmer: _____

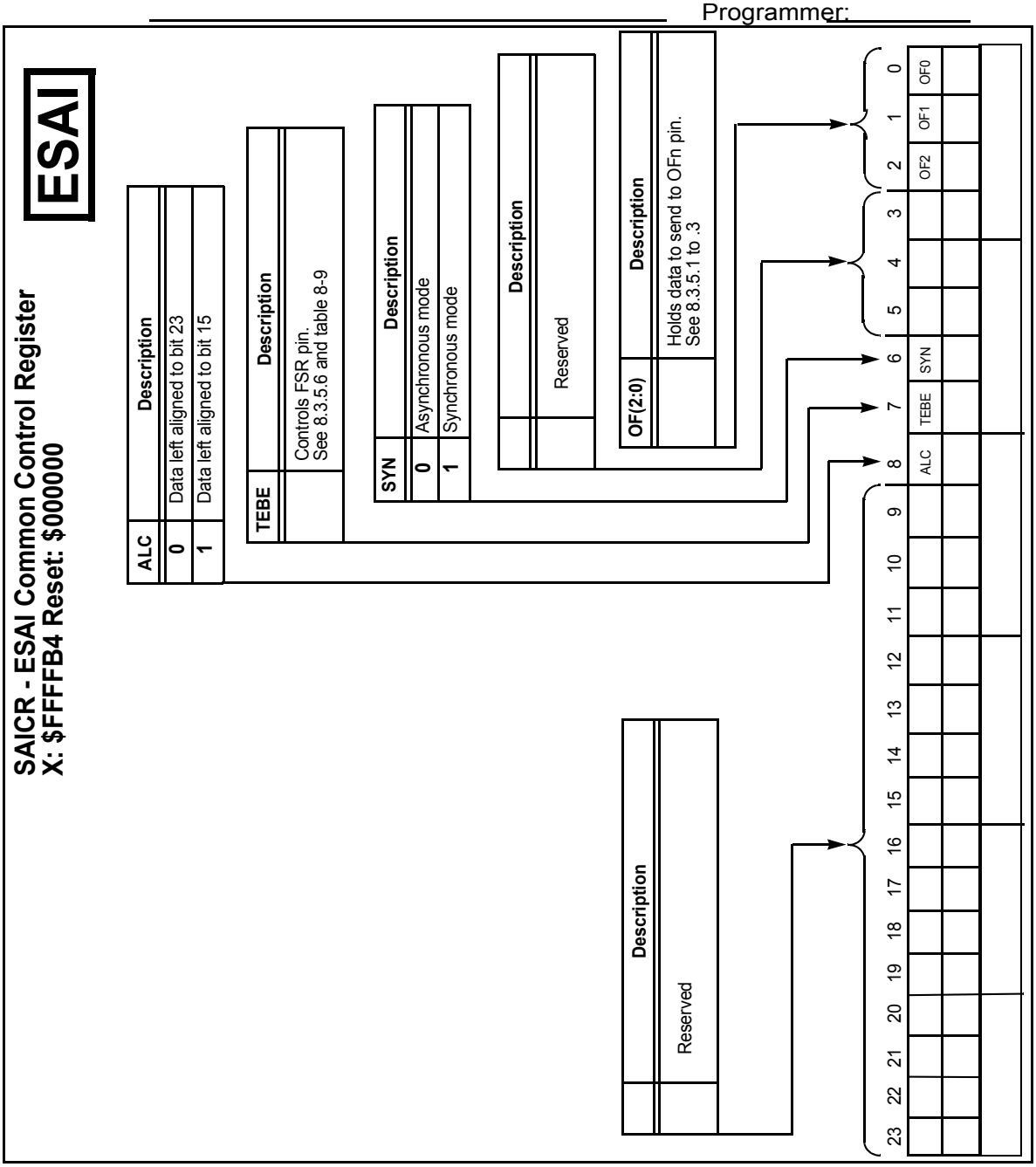


Figure D-19. ESAI Common Control Register

Application: _____ Date: _____

Programmer: _____

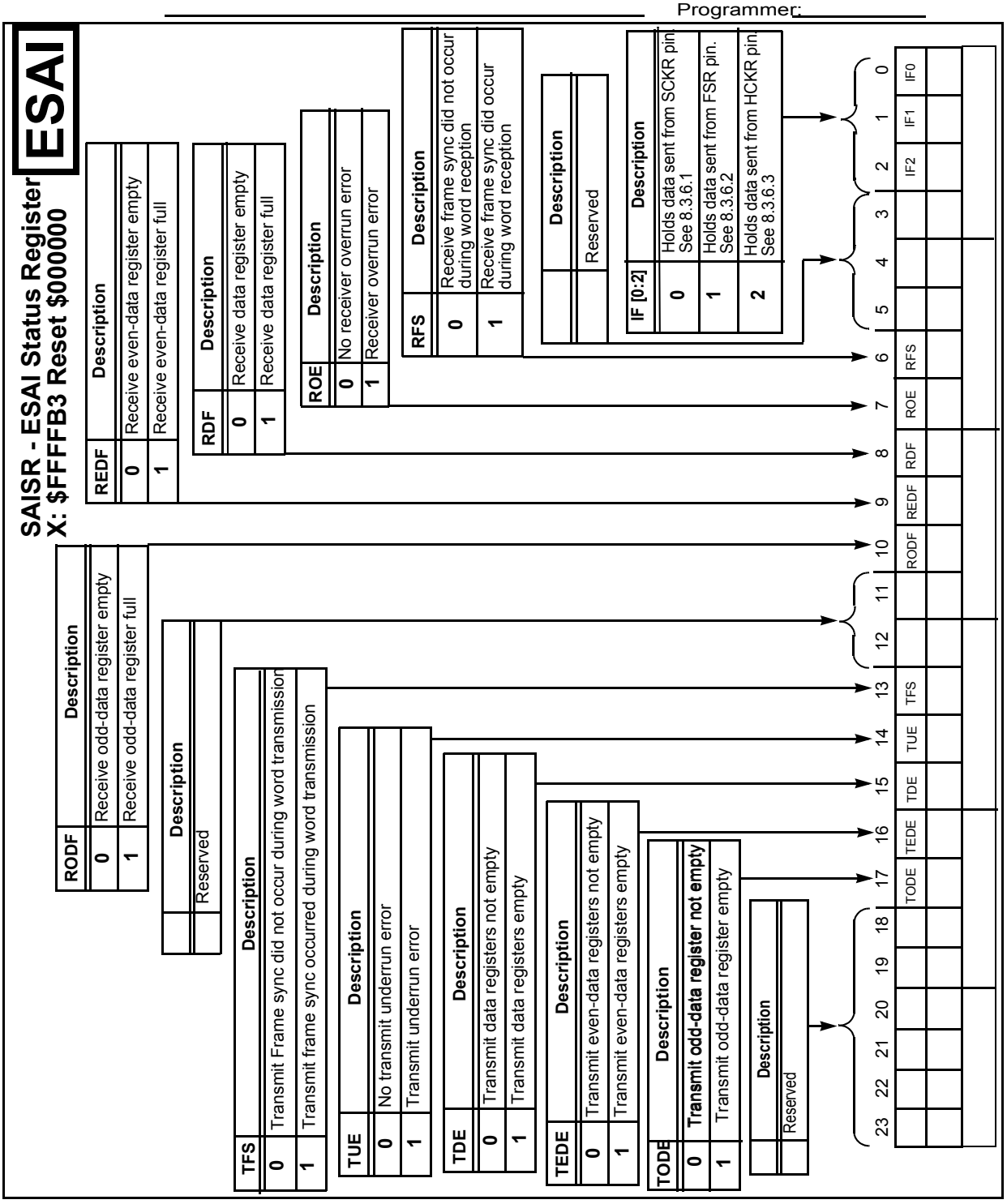

SAISR - ESAI Status Register
X: $FFFFB3 Reset $000000
ESAI
RODF
Description
0 Receive odd-data register empty
1 Receive odd-data register full
REDF
Description
0 Receive even-data register empty
1 Receive even-data register full
RDF
Description
0 Receive data register empty
1 Receive data register full
ROE
Description
0 No receiver overrun error
1 Receiver overrun error
RFS
Description
0 Receive frame sync did not occur during word reception
1 Receive frame sync did occur during word reception
IF [0:2]
Description
0 Holds data sent from SCKR pin See 8.3.6.1
1 Holds data sent from FSR pin. See 8.3.6.2
2 Holds data sent from HCKR pin See 8.3.6.3
Description
Reserved
TFS
Description
0 Transmit Frame sync did not occur during word transmission
1 Transmit frame sync occurred during word transmission
TUE
Description
0 No transmit underrun error
1 Transmit underrun error
TDE
Description
0 Transmit data registers not empty
1 Transmit data registers empty
TEDE
Description
0 Transmit even-data registers not empty
1 Transmit even-data registers empty
TODE
Description
0 Transmit odd-data register not empty
1 Transmit odd-data register empty
Description
Reserved
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
IF0 IF1 IF2 RFS ROE RDF REDF RODF TFS TUE TDE TEDE TODE

Figure D-20. ESAI Status Register

Application: _____ Date: _____

_____ Programmer: _____

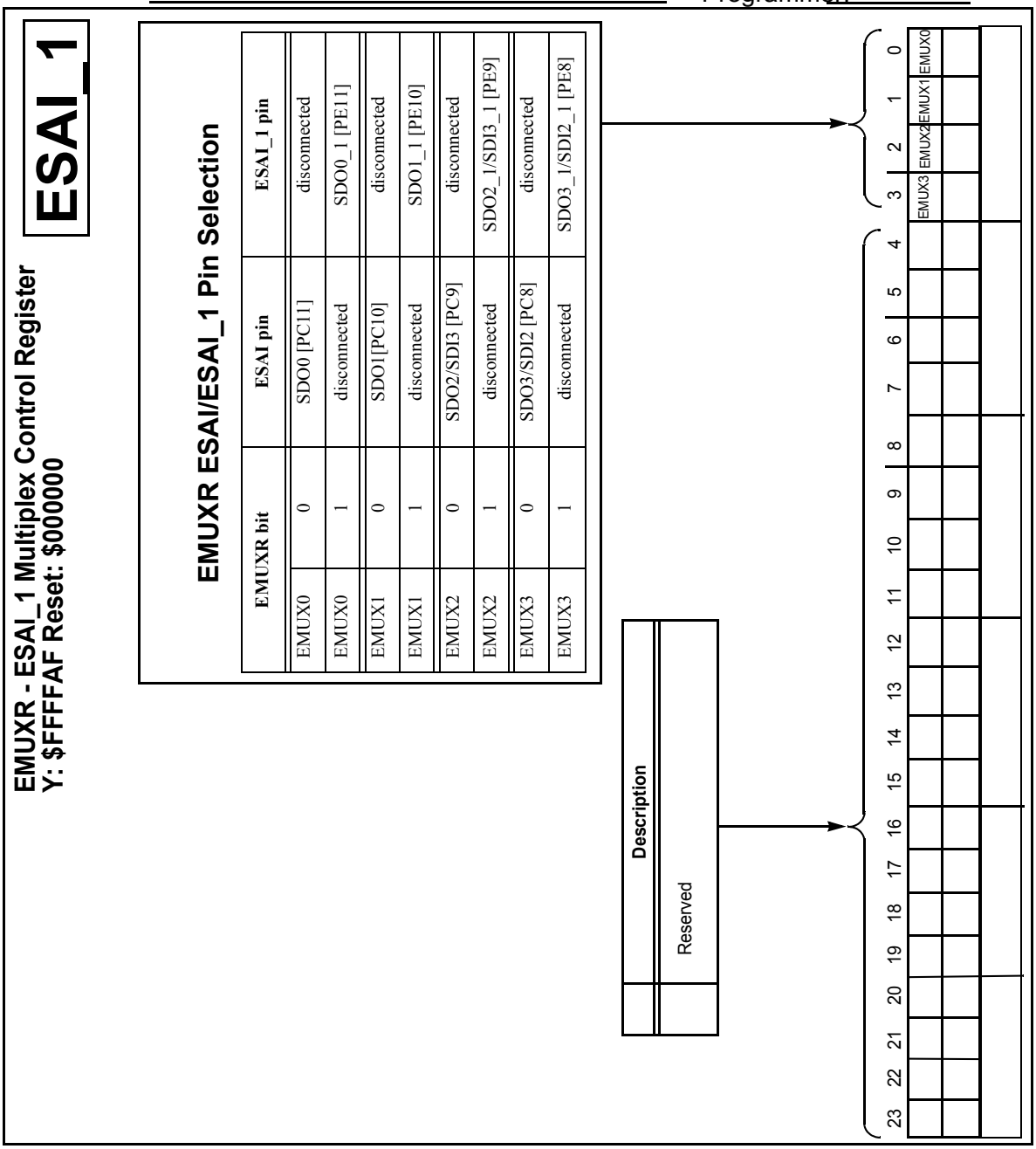

ESAI_1
EMUXR - ESAI_1 Multiplex Control Register
Y: $FFFFAF Reset: $000000
EMUXR ESAI/ESAI_1 Pin Selection
EMUXR bit
ESAI pin
ESAI_1 pin
EMUX0 0 SDO0 [PC11] disconnected
EMUX0 1 disconnected SDO0_1 [PE11]
EMUX1 0 SDO1 [PC10] disconnected
EMUX1 1 disconnected SDO1_1 [PE10]
EMUX2 0 SDO2/SDI3 [PC9] disconnected
EMUX2 1 disconnected SDO2_1/SDI3_1 [PE9]
EMUX3 0 SDO3/SDI2 [PC8] disconnected
EMUX3 1 disconnected SDO3_1/SDI2_1 [PE8]
Description
Reserved
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
EMUX3 EMUX2 EMUX1 EMUX0

Figure D-21. ESAI_1 Multiplex Control Register

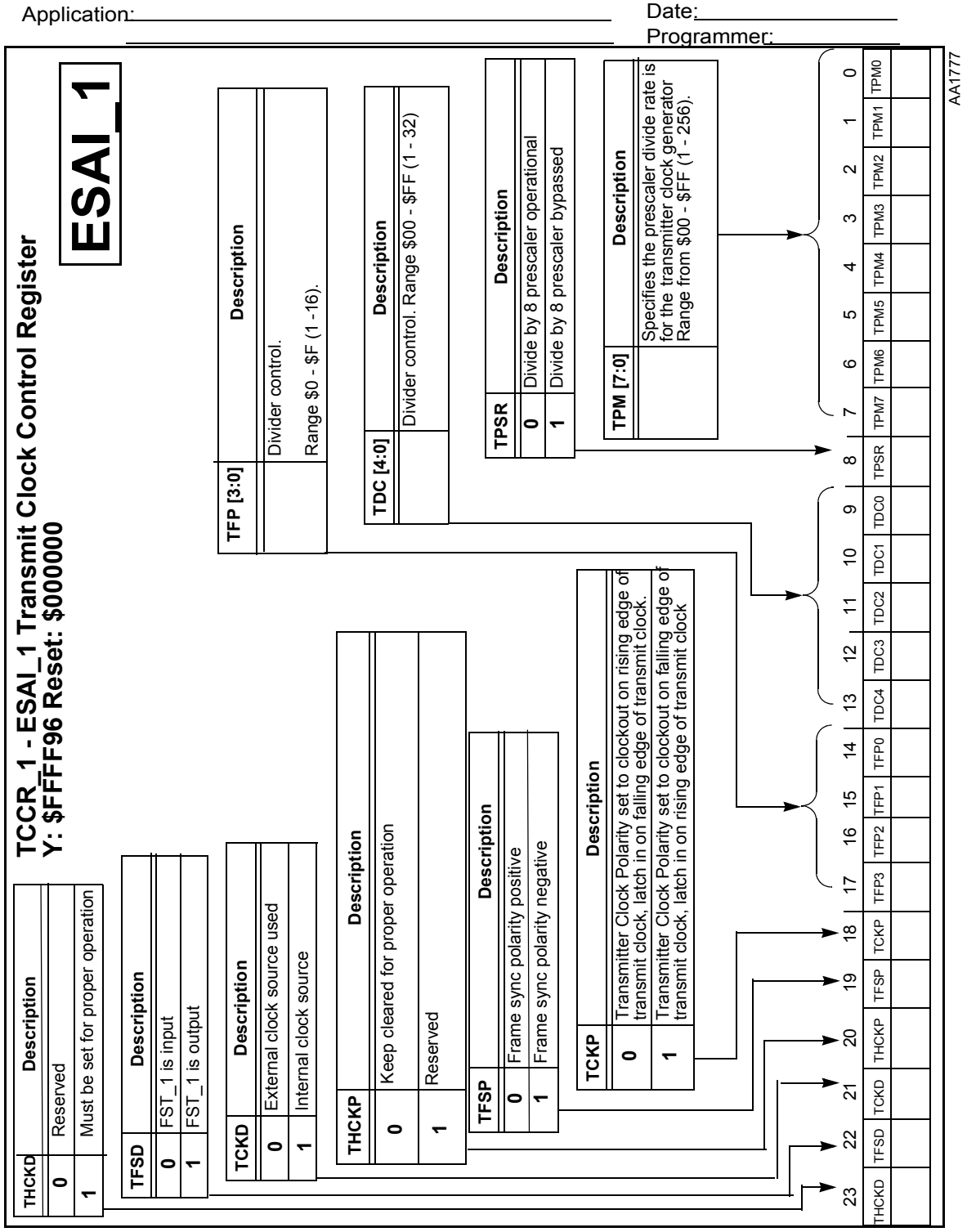


Figure D-22. ESAI_1 Transmit Clock Control Register

Application: _____ Date: _____

_____ Programmer: _____

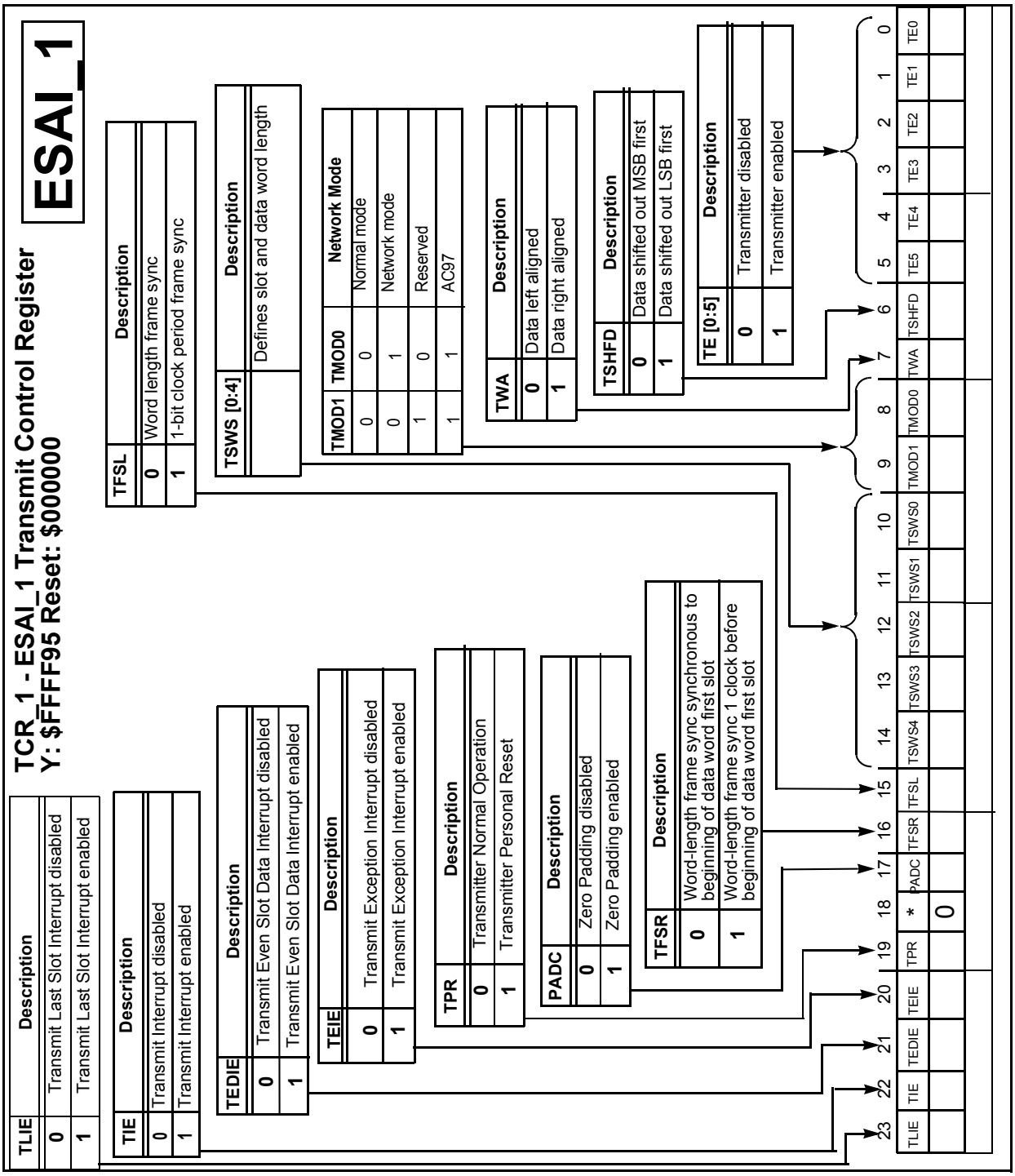

ESAI_1
TCR_1 - ESAI_1 Transmit Control Register
Y: $FFFF95 Reset: $000000
TLIE Description
0 Transmit Last Slot Interrupt disabled
1 Transmit Last Slot Interrupt enabled
TIE Description
0 Transmit Interrupt disabled
1 Transmit Interrupt enabled
TEDIE Description
0 Transmit Even Slot Data Interrupt disabled
1 Transmit Even Slot Data Interrupt enabled
TEIE Description
0 Transmit Exception Interrupt disabled
1 Transmit Exception Interrupt enabled
TPR Description
0 Transmitter Normal Operation
1 Transmitter Personal Reset
PADC Description
0 Zero Padding disabled
1 Zero Padding enabled
TFSR Description
0 Word-length frame sync synchronous to beginning of data word first slot
1 Word-length frame sync 1 clock before beginning of data word first slot
TFSL Description
0 Word length frame sync
1 1-bit clock period frame sync
TSWS [0:4] Description
Defines slot and data word length
TMOD1 TMOD0 Network Mode
0 0 Normal mode
0 1 Network mode
1 0 Reserved
1 1 AC97
TWA Description
0 Data left aligned
1 Data right aligned
TSHFD Description
0 Data shifted out MSB first
1 Data shifted out LSB first
TE [0:5] Description
0 Transmitter disabled
1 Transmitter enabled
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
TLIE TIE TEDIE TEIE TPR * PADC TFSR TFSL TSWS4 TSWS3 TSWS2 TSWS1 TSWS0 TMOD1 TMOD0 TWA TSHFD TE5 TE4 TE3 TE2 TE1 TE0
0

Figure D-23. ESAI_1 Transmit Control Register

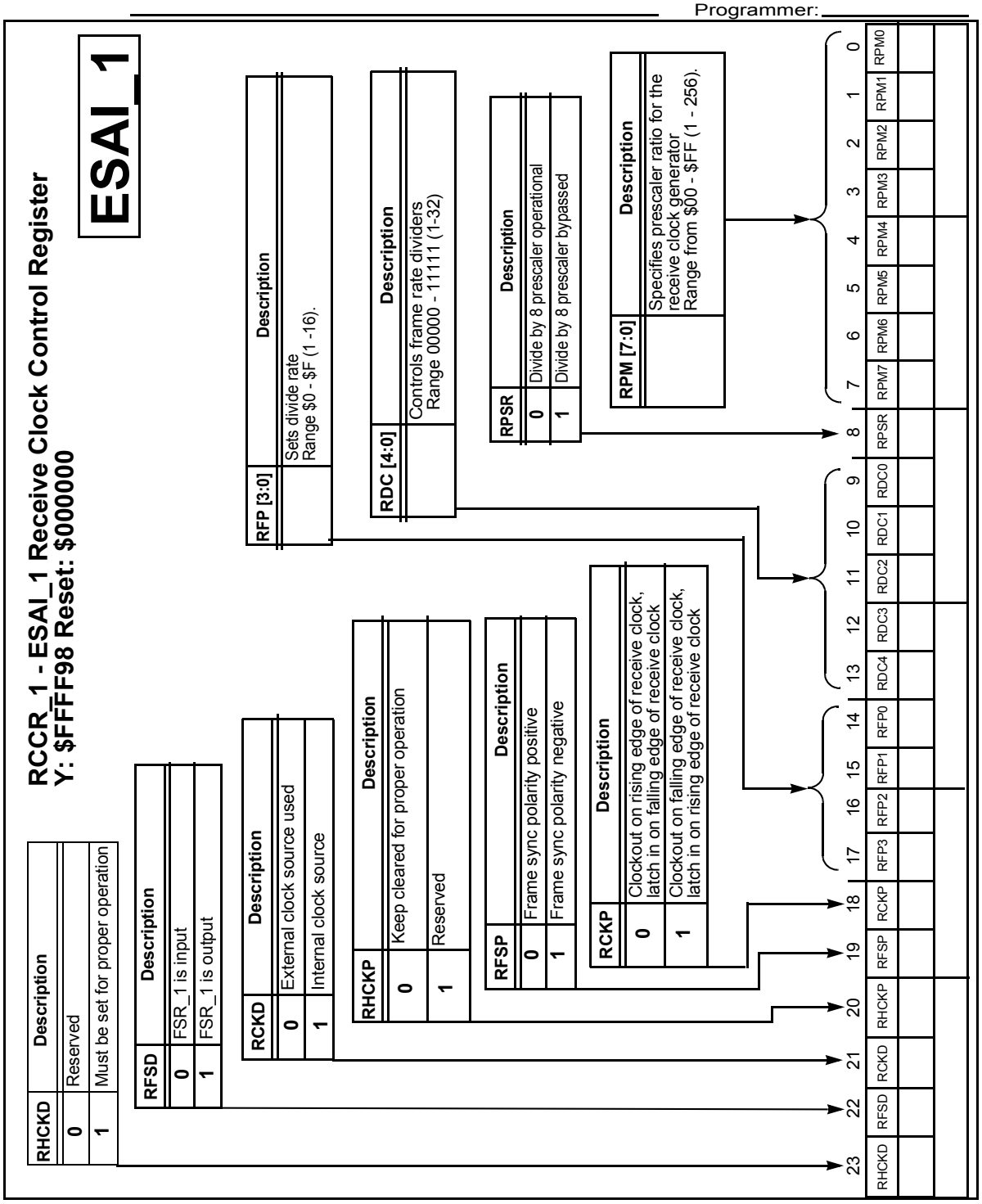


Figure D-24. ESAI_1 Receive Clock Control Register

Application: _____ Date: _____

Programmer: _____

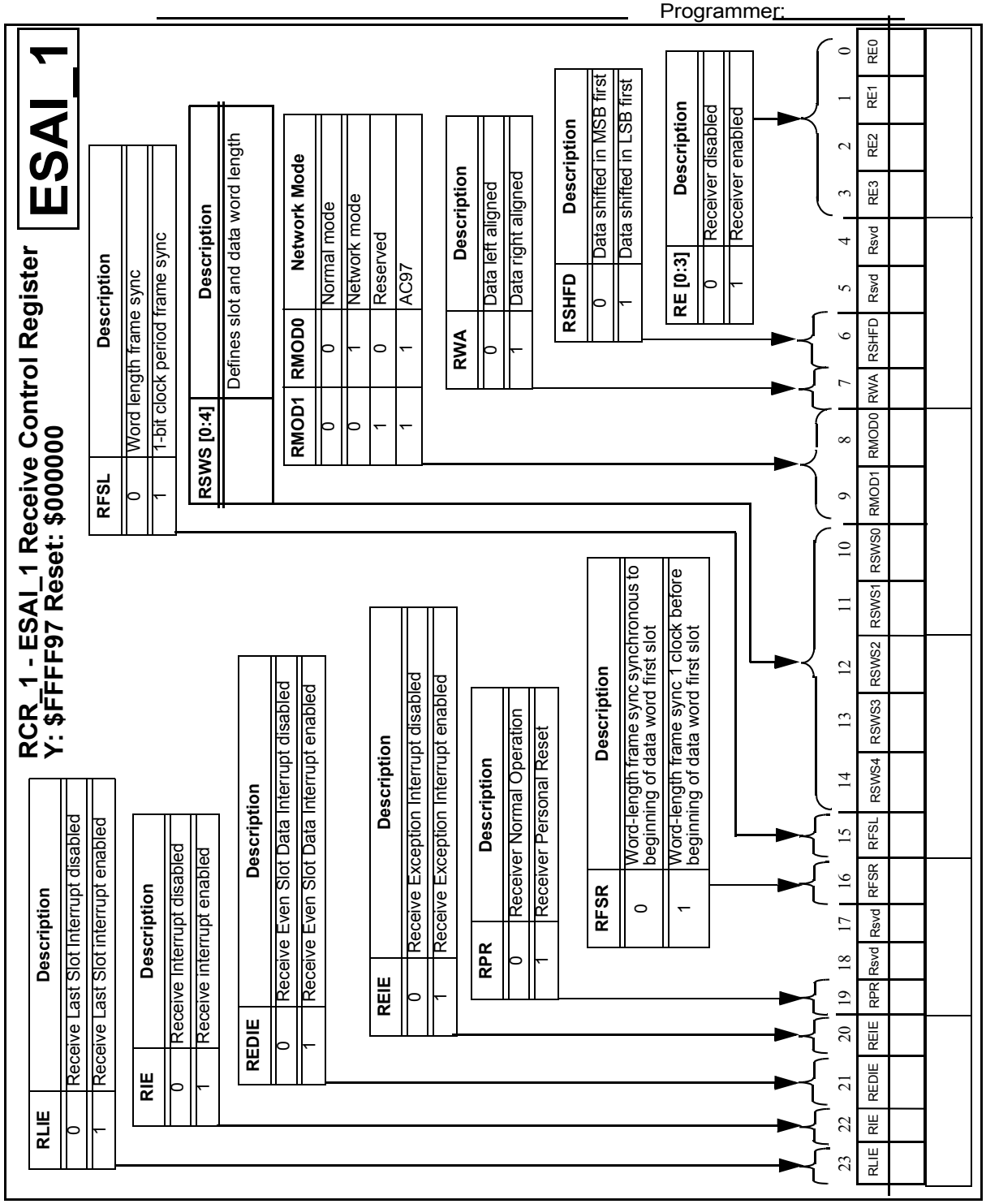


Figure D-25. ESAI_1 Receive Control Register

Application: _____ Date: _____

Programmer: _____

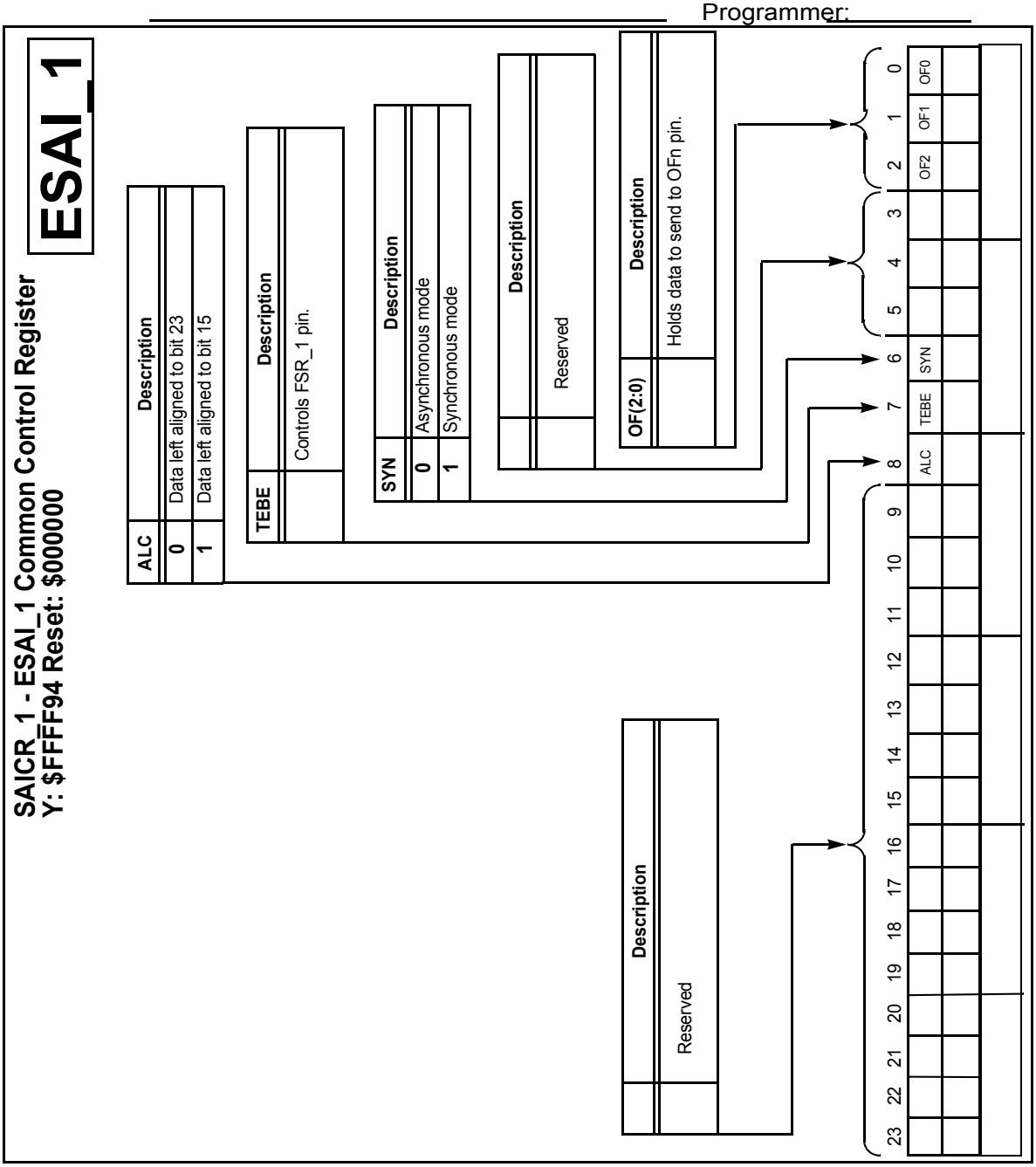


Figure D-26. ESAI_1 Common Control Register

Application: Date:

Programmer:

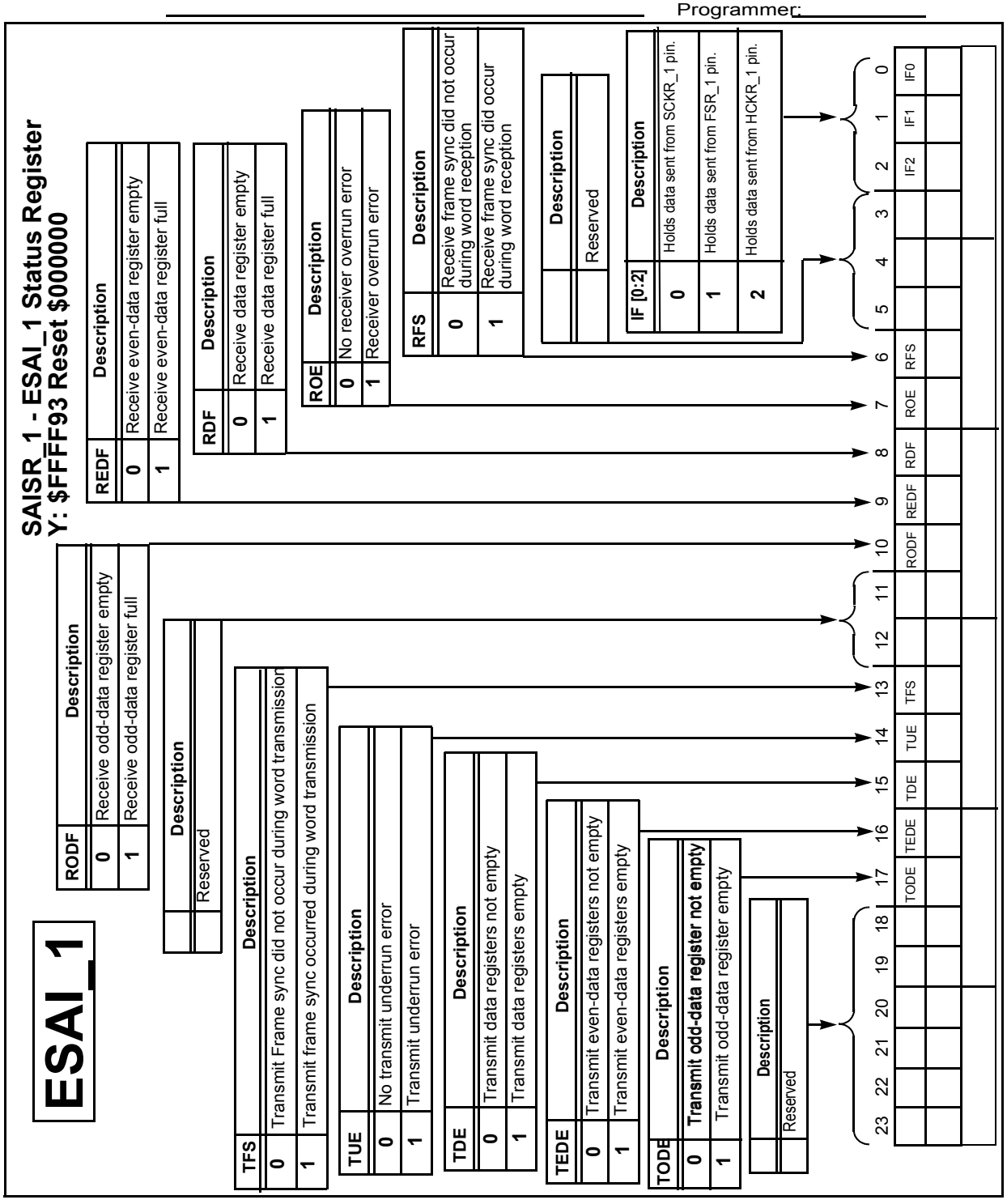


Figure D-27. ESAI_1 Status Register

Application:_____ Date:_____

_____ Programmer:_____

Sheet 1 of 2

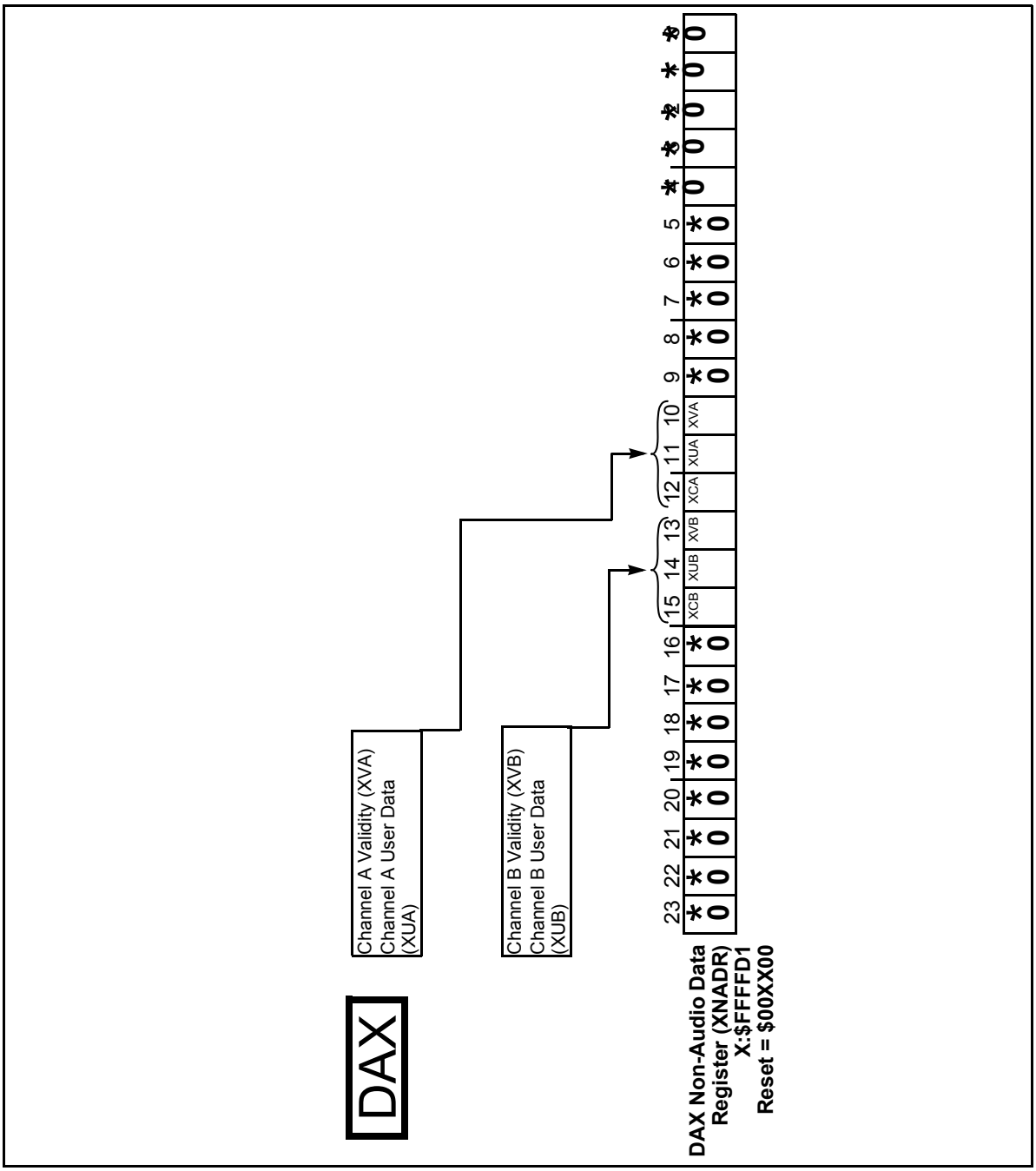

DAX
Channel A Validity (XVA)
Channel A User Data
(XUA)
Channel B Validity (XVB)
Channel B User Data
(XUB)
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
* 0
XCB XUB XVB XCA XUA XVA
DAX Non-Audio Data
Register (XNADR)
X:$FFFFD1
Reset = $00XX00

Figure D-28. DAX Non-Audio Data Register

Application: _____ Date: _____

_____ Programmer: _____

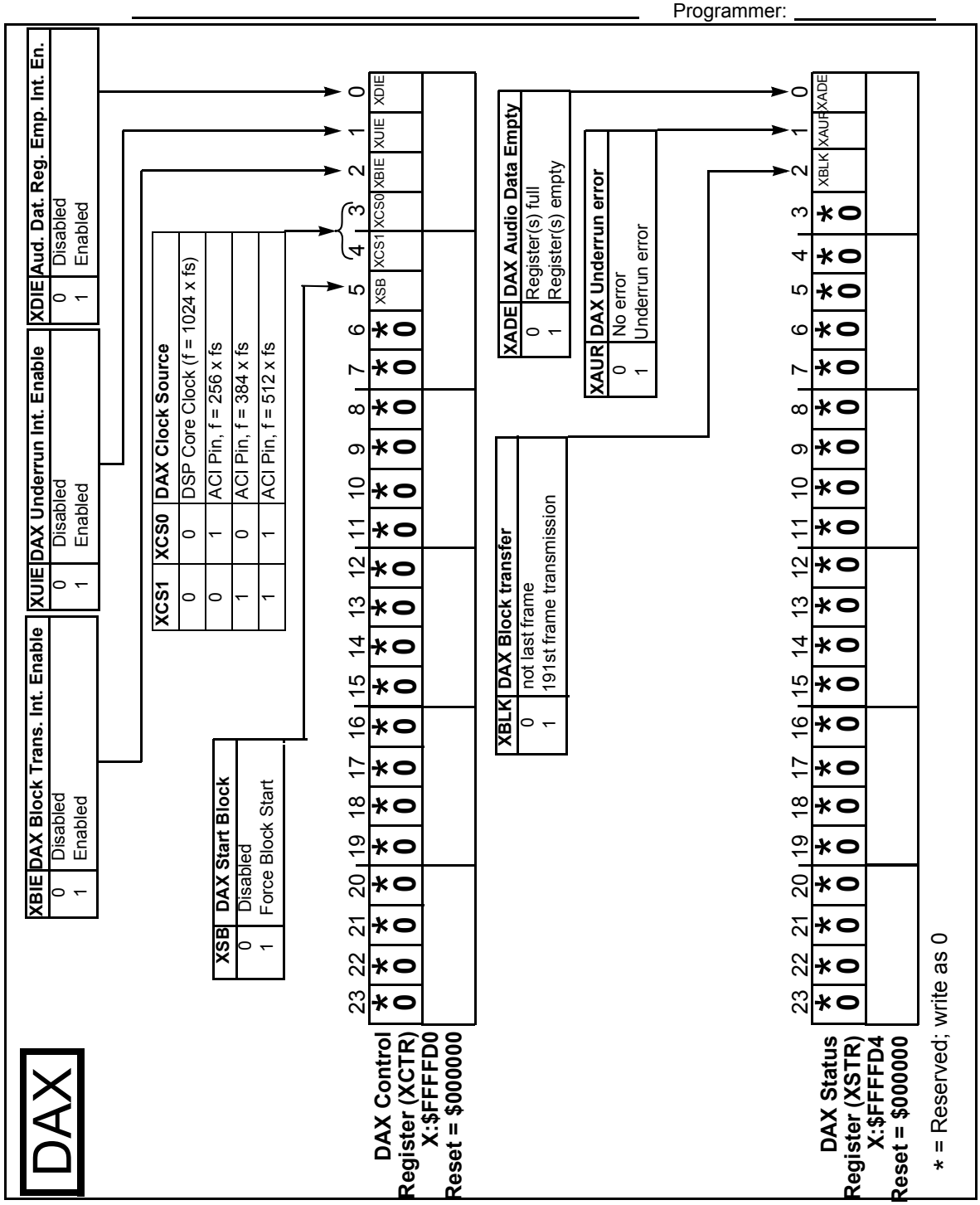

DAX
XBIE DAX Block Trans. Int. Enable
0 Disabled
1 Enabled
XUIE DAX Underrun Int. Enable
0 Disabled
1 Enabled
XDIE Aud. Dat. Reg. Emp. Int. En.
0 Disabled
1 Enabled
XCS1 XCS0 DAX Clock Source
0 0 DSP Core Clock (f = 1024 x fs)
0 1 ACI Pin, f = 256 x fs
1 0 ACI Pin, f = 384 x fs
1 1 ACI Pin, f = 512 x fs
XSB DAX Start Block
0 Disabled
1 Force Block Start
XBLK DAX Block transfer
0 not last frame
1 191st frame transmission
XADE DAX Audio Data Empty
0 Register(s) full
1 Register(s) empty
XAUR DAX Underrun error
0 No error
1 Underrun error
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
* *
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
XSB XCS1 XCS0 XBIE XUIE XDIE
DAX Control Register (XCTR) X:$FFFFD0 Reset = $000000
XBLK XAUR XADE
DAX Status Register (XSTR) X:$FFFFD4 Reset = $000000
* = Reserved; write as 0

Figure D-29. DAX Control and Status Registers

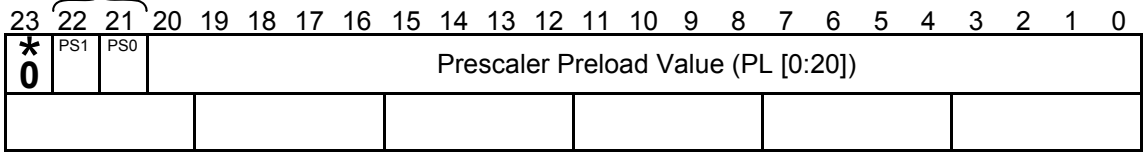
Application:_____ Date:_____

_____ Programmer:_____

Sheet 1 of 3

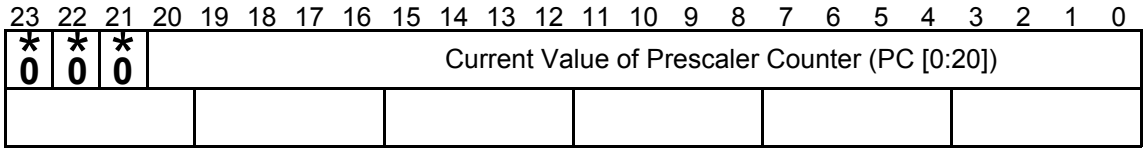
TEC

PS (1:0)	Prescaler Clock Source
00	Internal CLK/2
01	TIO0
10	Reserved
11	Reserved



Timer Prescaler Load Register
TPLR:$FFFF83 Read/Write
Reset = $000000

* = Reserved, Program as 0



Timer Prescaler Count Register
TPCR:$FFFF82 Read Only
Reset = $000000

*= Reserved, Program as 0

Figure D-30. Timer Prescaler Load and Prescaler Count Registers (TPLR, TPCR)

Application: _____ Date: _____

_____ Programmer: _____

Sheet 2 of 3

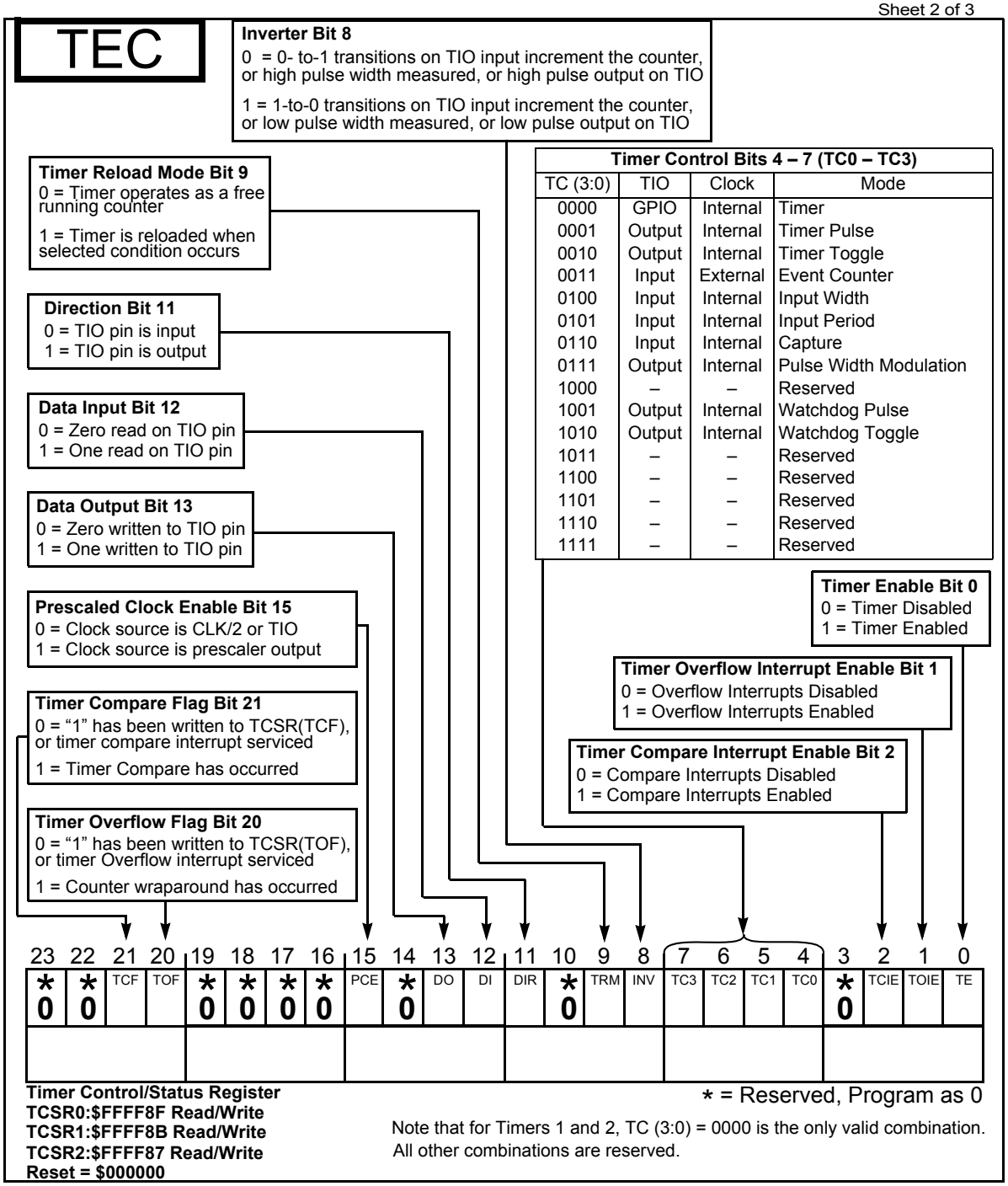

TEC
Inverter Bit 8
0 = 0- to-1 transitions on TIO input increment the counter, or high pulse width measured, or high pulse output on TIO
1 = 1-to-0 transitions on TIO input increment the counter, or low pulse width measured, or low pulse output on TIO
Timer Reload Mode Bit 9
0 = Timer operates as a free running counter
1 = Timer is reloaded when selected condition occurs
Direction Bit 11
0 = TIO pin is input
1 = TIO pin is output
Data Input Bit 12
0 = Zero read on TIO pin
1 = One read on TIO pin
Data Output Bit 13
0 = Zero written to TIO pin
1 = One written to TIO pin
Prescaled Clock Enable Bit 15
0 = Clock source is CLK/2 or TIO
1 = Clock source is prescaler output
Timer Compare Flag Bit 21
0 = "1" has been written to TCSR(TCF), or timer compare interrupt serviced
1 = Timer Compare has occurred
Timer Overflow Flag Bit 20
0 = "1" has been written to TCSR(TOF), or timer Overflow interrupt serviced
1 = Counter wraparound has occurred
Timer Control Bits 4 – 7 (TC0 – TC3)
TC (3:0) | TIO | Clock | Mode
0000 | GPIO | Internal | Timer
0001 | Output | Internal | Timer Pulse
0010 | Output | Internal | Timer Toggle
0011 | Input | External | Event Counter
0100 | Input | Internal | Input Width
0101 | Input | Internal | Input Period
0110 | Input | Internal | Capture
0111 | Output | Internal | Pulse Width Modulation
1000 | – | – | Reserved
1001 | Output | Internal | Watchdog Pulse
1010 | Output | Internal | Watchdog Toggle
1011 | – | – | Reserved
1100 | – | – | Reserved
1101 | – | – | Reserved
1110 | – | – | Reserved
1111 | – | – | Reserved
Timer Enable Bit 0
0 = Timer Disabled
1 = Timer Enabled
Timer Overflow Interrupt Enable Bit 1
0 = Overflow Interrupts Disabled
1 = Overflow Interrupts Enabled
Timer Compare Interrupt Enable Bit 2
0 = Compare Interrupts Disabled
1 = Compare Interrupts Enabled
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
* 0 * 0 TCF TOF * 0 * 0 * 0 * 0 PCE * 0 DO DI DIR * 0 TRM INV TC3 TC2 TC1 TC0 * 0 TCIE TOIE TE
Timer Control/Status Register
TCSR0:$FFFF8F Read/Write
TCSR1:$FFFF8B Read/Write
TCSR2:$FFFF87 Read/Write
Reset = $000000
* = Reserved, Program as 0
Note that for Timers 1 and 2, TC (3:0) = 0000 is the only valid combination.
All other combinations are reserved.

Figure D-31. Timer Control/Status Register

Application: _____ Date: _____

_____ Programmer: _____

Sheet 3 of 3

TEC

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Timer Reload Value																							

Timer Load Register
TLR0:$FFFF8E Write Only
TLR1:$FFFF8A Write Only
TLR2:$FFFF86 Write Only

Reset = $XXXXXX

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value Compared to Counter Value																							

Timer Compare Register
TCPR0:$FFFF8D Read/Write
TCPR1:$FFFF89 Read/Write
TCPR2:$FFFF85 Read/Write

Reset = $XXXXXX

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Timer Count Value																							

Timer Count Register
TCR0:$FFFF8C Read Only
TCR1:$FFFF88 Read Only
TCR2:$FFFF84 Read Only

Reset = $000000

Figure D-32. Timer Load, Compare and Count Registers

Application: _____ Date: _____

_____ Programmer: _____

Sheet 1 of 4

GPIO

Port B (HDI08)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Host Data Direction Register (HDDR)	DR15	DR14	DR13	DR12	DR11	DR10	DR9	DR8	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0
X:$FFFFC8 Read/Write																
Reset = $0																

DRx = 1 → PBx is Output DRx = 0 → PBx is Input

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Host Data Register (HDR)	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X:$FFFFC9 Read/Write																
Reset = Undefined																

Dx holds value of corresponding HDI08 GPIO pin.
Function depends on HDDR.

See the HDI08 HPCR Register (Figure D-8) for additional Port B GPIO control bits.

Figure D-33. GPIO Port B

Application: _____ Date: _____

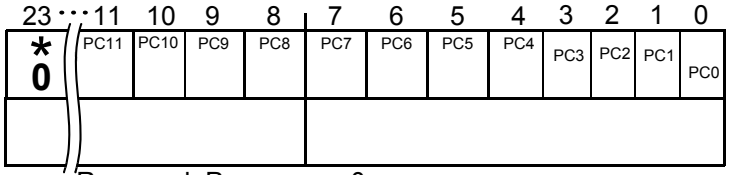
_____ Programmer: _____

Sheet 2 of 4

GPIO

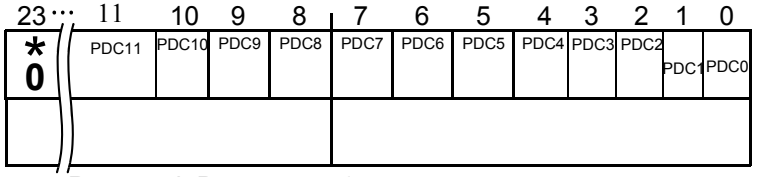
Port C (ESAI)

Port C Control Register
(PCRC)
X:$FFFFBF
Read/Write
Reset = $0



* = Reserved, Program as 0

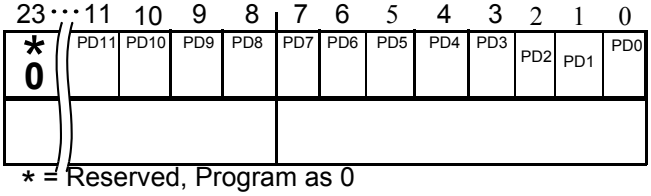
Port C Direction Register
(PRRC)
X:$FFFFBE
Read/Write
Reset = $0



* = Reserved, Program as 0

PCn = 0 & PDCn = 0 -> Port pin PCn disconnected
PCn = 1 & PDCn = 0 -> Port pin PCn configured as input
PCn = 0 & PDCn = 1 -> Port pin PCn configured as output
PCn = 1 & PDCn = 1 -> Port pin configured as ESAI

Port C GPIO Data Register
(PDRC)
X:$FFFFBD
Read/Write
Reset = undefined



* = Reserved, Program as 0

If port pin n is GPIO input, then PDn reflects the value on port pin n
if port pin n is GPIO output, then value written to PDn is reflected on port pin n

Figure D-34. GPIO Port C

Application: _____ Date: _____

_____ Programmer: _____

GPIO

Port D (DAX)

Port D Control Register (PCRD)
X:$FFFFD7
Read/Write
Reset = $0

23 ··· 6	5	4	3	2	1	0
* 0	* 0	* 0	* 0	* 0	PC1	PC0

* = Reserved, Program as 0

Port D Direction Register (PRRD)
X:$FFFFD6
Read/Write
Reset = $0

23 ··· 6	5	4	3	2	1	0
* 0	* 0	* 0	* 0	* 0	PDC1	PDC0

* = Reserved, Program as 0

PCn = 0 & PDCn = 0 -> Port pin PDn disconnected
PCn = 1 & PDCn = 0 -> Port pin PDn configured as input
PCn = 0 & PDCn = 1 -> Port pin PDn configured as output
PCn = 1 & PDCn = 1 -> Port pin configured as DAX

Port D GPIO Data Register (PDRD)
X:$FFFFD5
Read/Write
Reset = $0

23 ··· 6	5	4	3	2	1	0
* 0	* 0	* 0	* 0	* 0	PD1	PD0

* = Reserved, Program as 0

If port pin n is GPIO input, then PDn reflects the value on port pin n

if port pin n is GPIO output, then value written to PDn is reflected on port pin n

Figure D-35. GPIO Port D

Application: _____ Date: _____

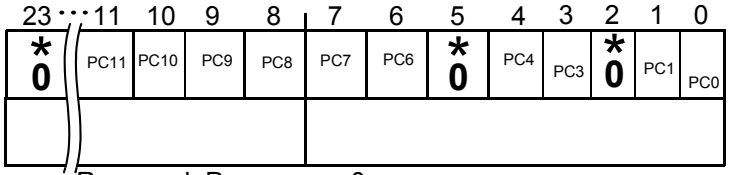
_____ Programmer: _____

Sheet 4 of 4

GPIO

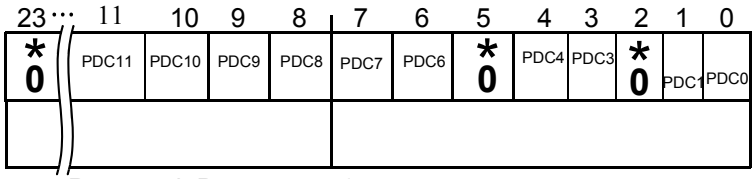
Port E (ESAI_1)

Port E Control Register
(PCRE)
Y:$FFFF9F
Read/Write
Reset = $0



* = Reserved, Program as 0

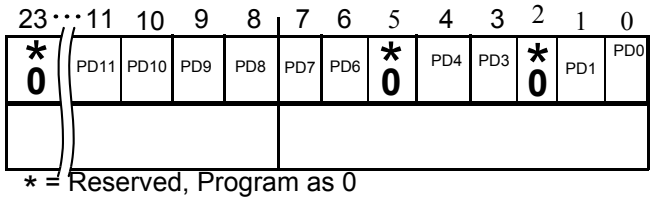
Port E Direction Register
(PRRE)
Y:$FFFF9E
Read/Write
Reset = $0



* = Reserved, Program as 0

PCn = 0 & PDCn = 0 -> Port pin PEn disconnected
PCn = 1 & PDCn = 0 -> Port pin PEn configured as input
PCn = 0 & PDCn = 1 -> Port pin PEn configured as output
PCn = 1 & PDCn = 1 -> Port pin configured as ESAI_1

Port E GPIO Data Register
(PDRE)
Y:$FFFF9D
Read/Write
Reset = undefined



* = Reserved, Program as 0

If port pin n is GPIO input, then PDn reflects the value on port pin n
if port pin n is GPIO output, then value written to PDn is reflected on port pin n

Figure D-36. GPIO Port E

NOTES

Index

Numerics

5 V tolerance 1

A

adder
    modulo 5
    offset 5
    reverse-carry 5
address bus 1
Address Generation Unit 5
addressing modes 5
AES/EBU 10, 1
AGU 5

B

barrel shifter 4
bus
    external address 5
    external data 5
bus control 1
buses
    internal 6

C

Central Processing Unit (CPU) i
CLKGEN 7
Clock 4
clock 1
Clock divider 11
Clock Generator (CLKGEN) 7
CP-340 10, 1
CPHA and CPOL (HCKR Clock Phase and Polarity Controls) 7

D

data ALU 4
    registers 4
data bus 1
Data Output bit (DO) 10
DAX 1, 21
    Block Transferred Interrupt Handling 11
    Initiating A Transmit Session 10
    Transmit Register Empty Interrupt Handling 10
DAX Audio Data register Empty (XADE) status flag 8
DAX Audio Data Registers (XADRA/XADRB) 5
DAX Audio Data Shift Register (XADSR) 5
DAX biphase encoder 9
DAX Block transfer (XBLK) flag 8
DAX Channel A Channel status (XCA) bit 6
DAX Channel A User data (XUA) bit 6
DAX Channel A Validity (XVA) bit 5
DAX Channel B Channel Status (XCB) bit 6
DAX Channel B User Data (XUB) bit 6
DAX Channel B Validity (XVB) bit 6
DAX Clock input Select bits 7
DAX clock multiplexer 9
DAX clock selection 7
DAX Control Register (XCTR) 6
DAX internal architecture 4
DAX Interrupt Enable (XIEN) bit 7
DAX Non-Audio Data Buffer (XNADBUF) 6
DAX Operation During Stop 12
DAX Parity Generator (PRTYG) 9
DAX preamble generator 9
DAX Preamble sequence 9, 12
DAX Programming Considerations 10
DAX programming model 3
DAX Status Register (XSTR) 7
DAX Transmit Underrun error (XAUR) status flag 8
DI 10
Digital Audio Transmitter 1, 21
Digital Audio Transmitter (DAX) 10, 1
DIR 10
Divide Factor (DF) 7
DMA 6

triggered by timer 21
DO bit 10
DO loop 6
DRAM 8
DSP56300 core 2
DSP56300 Family Manual i, 2
DSP56303 Technical Data i

E

Enhanced Serial Audio Interface 15, 19
Enhanced Synchronous Audio Interface 1
ESAI 1, 15, 19
ESAI block diagram 1
ESSI0 (GPIO) 1, 2
ESSI1 (GPIO) 1
external address bus 5
external bus control 5, 6, 7
external data bus 5
External Memory Expansion Port 5

F

functional signal groups 1

G

Global Data Bus 6
GPIO 9, 22
GPIO (ESSI0, Port C) 1, 2
GPIO (ESSI1, Port D) 1
GPIO (HI08, Port B) 1
GPIO (Timer) 2
Ground 3
ground 1

H

HA1, HA3-HA6 (HSAR I^2C Slave Address) 7
hardware stack 5
HBER (HCSR Bus Error) 16
HBIE (HCSR Bus Error Interrupt Enable) 13
HBUSY (HCSR Host Busy) 16
HCKR (SHI Clock Control Register) 7
HCSR
Receive Interrupt Enable Bits 14
SHI Control/Status Register 10
HDI08 1, 9, 11, 12
HDM0-HDM5 (HCKR Divider Modulus Select) 9
HEN (HCSR SHI Enable) 10
HFIFO (HCSR FIFO Enable Control) 12
HFM0-HFM1 (HCKR Filter Mode) 9
HI08 9
(GPIO) 1
HI^{2}C (HCSR Serial Host Interface I^2C/SPI Selection) 11
HIDLE (HCSR Idle) 13
HM0-HM1 (HCSR Serial Host Interface Mode) 11
HMST (HCSR Master Mode) 12
Host
Receive Data FIFO (HRX) 6
Receive Data FIFO—DSP Side 6
Transmit Data Register (HTX) 6
Transmit Data Register—DSP Side 6
Host Interface 9, 1, 9, 11, 12
HREQ Function In SHI Slave Modes 12
HRFF (HCSR Host Receive FIFO Full) 15
HRIE0-HRIE1 (HCSR Receive Interrupt Enable) 14
HRNE (HCSR Host Receive FIFO Not Empty) 15
HROE (HCSR Host Receive Overrun Error) 15
HRQE0-HRQE1 (HCSR Host Request Enable) 12
HTDE (HCSR Host Transmit Data Empty) 15
HTIE (HCSR Transmit Interrupt Enable) 13
HTUE (HCSR Host Transmit Underrun Error) 14

I

I^2C 10, 1, 16
Bit Transfer 17
Bus Protocol For Host Read Cycle 19
Bus Protocol For Host Write Cycle 18
Data Transfer Formats 18
Master Mode 23
Protocol for Host Write Cycle 18
Receive Data In Master Mode 24
Receive Data In Slave Mode 21
Slave Mode 21
Start and Stop Events 17

Transmit Data In Master Mode 24
Transmit Data In Slave Mode 22
I2C Bus Acknowledgment 18
I2C Mode 1
IEC958 10, 1
Inter Integrated Circuit Bus 10, 1
internal buses 6
Internal Exception Priorities
SHI 5
interrupt 6
interrupt and mode control 1, 7, 8
interrupt control 7, 8
Interrupt Vectors
SHI 5
INV 9

J

JTAG 7, 22
JTAG/OnCE port 1

L

LA register 6
LC register 6
Loop Address register (LA) 6
Loop Counter register (LC) 6

M

MAC 4
Manual Conventions ii
memory
expansion 8
external expansion port 8
off-chip 8
on-chip 7
mode control 7, 8
modulo adder 5
multiplier-accumulator (MAC) 4

O

offset adder 5
OMR register 6
OnCE module 7, 22
On-Chip Emulation (OnCE) module 7
on-chip memory 7
Operating Mode Register (OMR) 6

P

PAB 6
PAG 5
PC register 6
PC0-PC20 bits 6
PCE 11
PCU 5
PDB 6
PDC 5
Peripheral I/O Expansion Bus 6
PIC 5
PL0-PL20 bits 5
PL21-PL22 bits 5
PLL 7, 1, 4
Port A 1, 5
Port B 1, 9, 10, 11, 12, 1
Port C 1, 15, 19, 1, 2
Port D 21, 1
Power 2
power 1
Prescaler Counter 5
Prescaler Counter Value bits (PC0-PC20) 6
Prescaler Load Value bits (PL0-PL20) 5
Prescaler Source bits (PL21-PL22) 5
Program Address Bus (PAB) 6
Program Address Generator (PAG) 5
Program Control Unit (PCU) 5
Program Counter register (PC) 6
Program Data Bus (PDB) 6
Program Decode Controller (PDC) 5
Program Interrupt Controller (PIC) 5
Program Memory Expansion Bus 6
Programming Model
SHI—DSP Side 4
SHI—Host Side 3

R

reserved bits
  in TCSR register
    bits 3, 10, 14, 16–19, 22, 23 11
  in TPCR 6
  in TPLR 6
RESET 8
reverse-carry adder 5

S

SC register 6
Serial Host Interface 1, 12
Serial Host Interface (SHI) 10, 1
Serial Peripheral Interface Bus 10, 1
SHI 10, 1, 12, 1
  Block Diagram 2
  Clock Control Register—DSP Side 7
  Clock Generator 2, 3
  Control/Status Register—DSP Side 10
  Data Size 11
  Exception Priorities 5
  HCKR
    Clock Phase and Polarity Controls 7
    Divider Modulus Select 9
    Prescaler Rate Select 9
  HCKR Filter Mode 9
  HCSR
    Bus Error Interrupt Enable 13
    FIFO Enable Control 12
    Host Request Enable 12
    Idle 13
    Master Mode 12
    Serial Host Interface I^2C/SPI Selection 11
    Serial Host Interface Mode 11
    SHI Enable 10
  Host Receive Data FIFO—DSP Side 6
  Host Transmit Data Register—DSP Side 6
  HREQ
    Function In SHI Slave Modes 12
  HSAR
    I^2C Slave Address 7
    Slave Address Register 6
  I/O Shift Register 6
  Input/Output Shift Register—Host Side 5
  Internal Architecture 2
  Internal Interrupt Priorities 5
  Interrupt Vectors 5
  Introduction 1
  Operation During Stop 25
  Programming Considerations 19
  Programming Model 3
  Programming Model—DSP Side 4
  Programming Model—Host Side 3
  Slave Address Register—DSP Side 6
SHI Noise Reduction Filter Mode 10
signal groupings 1
signals 1
Size register (SZ) 6
SP 6
SPI 10, 1
  HCSR
    Bus Error 16
    Host Busy 16
    Host Receive FIFO Full 15
    Host Receive FIFO Not Empty 15
    Host Receive Overrun Error 15
    Host Transmit Data Empty 15
    Host Transmit Underrun Error 14
    Receive Interrupt Enable 13, 14
  Master Mode 20
  Slave Mode 19
SPI Data-To-Clock Timing 8
SPI Data-To-Clock Timing Diagram 8
SPI Mode 1
SR register 6
SRAM
  interfacing 8
SS 6
Stack Counter register (SC) 6
Stack Pointer (SP) 6
Status Register (SR) 6
System Stack (SS) 6
SZ register 6

T

TAP 7

TC0–TC3 bits 7
TCF 11
TCIE bit 7
TCPR 12
TCR 12
TCSR register 6
  bit 0—Timer Enable bit (TE) 6
  bit 2—Timer Compare Interrupt Enable bit (TCIE) 7
  bits 4–7—Timer Control bits (TC0–TC3) 7
  bit 13—Data Output bit (DO) 10
  reserved bits—bits 3, 10, 14, 16–19, 22, 23 11
TE bit 6
Test Access Port (TAP) 7
Timer 1, 22
timer
  special cases 20
Timer (GPIO) 2
Timer Compare Interrupt Enable bit (TCIE) 7
Timer Control bits (TC0–TC3) 7
Timer Control/Status Register (TCSR) 6
Timer Enable bit (TE) 6
timer mode
  mode 0—GPIO 13
  mode 1—timer pulse 14
  mode 2—timer toggle 14
  mode 3—timer event counter 15
  mode 4—measurement input width 16
  mode 5—measurement input period 17
  mode 6—measurement capture 17
  mode 7—pulse width modulation 18
  mode 8—reserved 19
  mode 9—watchdog pulse 19, 20
  modes 11–15—reserved 20
Timer module
  architecture 1
Timer Prescaler Count Register (TPCR) 6
Timer Prescaler Load Register (TPLR) 5
TLR 12
TOF 11
TOIE 7
TPCR register 6
  bits 0-20—Prescaler Counter Value bits (PC0-PC20) 6
  bit 21-23—reserved bits 6
  reserved bits—bits 21-23 6
TPLR register 5
  bits 0-20—Prescaler Load Value bits (PL0-PL20) 5
  bits 21-22—Prescaler Source bits (PL0-PL20) 5
  bit 23—reserved bit 6
  reserved bit—bit 23 6
Transmitter High Frequency Clock Divider 11
TRM 10

V

VBA register 6
Vector Base Address register (VBA) 6

X

X Memory Address Bus (XAB) 6
X Memory Data Bus (XDB) 6
X Memory Expansion Bus 6
XAB 6
XDB 6

Y

Y Memory Address Bus (YAB) 6
Y Memory Data Bus (YDB) 6
Y Memory Expansion Bus 6
YAB 6
YDB 6